Universidade do Algarve

Instituto Superior de Engenharia

Visão Activa para Robô Cognitivo

Tese de Mestrado em Engenharia Eléctrica e Electrónica,
Área de Especialização: Tecnologias de Informação e Telecomunicações

Mário Alexandre Nobre Saleiro

Dissertação orientada pelo Professor Doutor João Rodrigues

Faro

(Março, 2011)

Universidade do Algarve

Instituto Superior de Engenharia

Visão Activa para Robô Cognitivo

Tese de Mestrado em Engenharia Eléctrica e Electrónica,
Área de Especialização: Tecnologias de Informação e Telecomunicações

Mário Alexandre Nobre Saleiro

Faro

(Março, 2011)

Agradecimentos

Em primeiro lugar gostaria de agradecer aos meus pais por sempre me terem apoiado, por me terem proporcionado o ambiente, motivação e bens necessários para chegar até aqui, por me ensinarem a definir e lutar pelos meus objectivos e por me transmitirem todos os valores morais que fizeram de mim a pessoa que sou hoje. Gostaria também de agradecer ao meu irmão por me ter ensinado muitas coisas, tais como programação quando ainda andava na escola primária, despertando assim o interesse pelo caminho que sigo hoje. À minha família, um obrigado muito especial.

Em segundo gostaria de fazer um agradecimento especial ao meu orientador, Prof. Doutor João Rodrigues que me ensinou tudo o que sei hoje sobre Visão Computacional e me fez aprender muito sobre outras áreas associadas. Obrigado também pela disponibilidade e pela prontidão no acompanhamento deste trabalho e não só. Agradeço também a todos os docentes do DEE do ISE que me transmitiram todos os conhecimentos e ferramentas para aprender cada vez mais e vir a ter sucesso na minha vida profissional.

Por fim, mas não com menor significado, obrigado a todos os amigos que me acompanharam e apoiaram durante a minha vida académica e nos últimos tempos tantas vezes me perguntaram "então, como vai essa tese?". Fábio, Gonçalo, Palma, Betty, Irina, Ana e Maria, especialmente para vocês um muito obrigado por tudo.

NOME: Mário Alexandre Nobre Saleiro

INSTITUIÇÃO: Instituto Superior de Engenharia

ORIENTADOR: Professor Doutor João Miguel Fernandes Rodrigues

DATA: Março de 2011

TITULO DA TESE: Visão Activa para Robô Cognitivo

Resumo

O interesse na robótica cognitiva tem vindo a aumentar ao longo dos tempos, tendo-se como grande objectivo a tentativa de criar uma estrutura que permita aos robôs adaptarem-se a ambientes dinâmicos e que permita a existência de uma aprendizagem por parte do robô a partir das suas experiências. Nesta dissertação é apresentada uma arquitectura para um robô cognitivo que segue o modelo cognitivo humano. A arquitectura desenvolvida combina diversas componentes, tais como estruturas de memória, sistema visual e sistema de aprendizagem. Cada uma destas componentes foi desenvolvida tendo como modelo os sistemas correspondentes do ser humano. No que diz respeito à memória, é feita a distinção entre memória a curto e a longo prazo como forma de seleccionar a informação que deve ser guardada e que informação deve ser desprezada. O sistema visual implementado faz uso da saliência e do foco-de-atenção como formas de seleccionar que informação pode ser prioritariamente processada nas imagens captadas. A informação seleccionada é depois processada para se efectuar o reconhecimento de objectos que podem servir como pontos de referência para que o robô se possa localizar. O sistema de aprendizagem permite ao robô construir ciclos visio-motores de acção complexos a partir de outros ciclos visio-motores mais simples que tenham sido previamente "ensinados" ao robô, fazendo assim uso das experiências passadas para realizar tarefas mais complexas, através da combinação de tarefas mais simples. Juntamente com esta arquitectura apresenta-se ainda um sistema de localização e mapeamento simultâneo que permite ao robô selectivamente explorar e mapear ambientes desconhecidos, guardando essa informação nas suas memórias a curto e a longo prazo.

PALAVRAS-CHAVE: Robótica; SLAM; Cognição; Mapeamento; Memória; Atenção; Saliência; Foco-de-atenção; Reconhecimento de objectos.

Active Vision for Cognitive Robot

Abstract

The interest in cognitive robotics has been increasing over time and the major goal is the attempt to create a cognitive structure that enables robots do adapt to dynamic environments and to learn with their own experiences. In this thesis a cognitive architecture for robots which follosws the human cognitive model as an example is presented. The developed architecture combines several components such as memory structures, a visual system and an apprenticeship system which were modelled after the corresponding human systems. The memory is split into short term memory and long term memory, allowing the robot to select and manage which information should be stored and which information should be disregarded. The visual system combines saliency and Focus-of-Attention as a way to select which information should be primarily processed in captured images. The selected information is then processed for object recognition creating reference points that can be used by the robot to locate itself. The learning system allows the robot to build complex cycles of visio-motor actions from simpler cycles that have been previously "taught" to the robot. This allows the robot to make use of past experiences to perform more complex tasks through the combination of simpler tasks. Along with this architecture a simultaneous localisation and mapping system is presented. This system allows the robot to selectively explore and map unknown environments, keeping the necessary information in the short and long term memories.

KEYWORDS: Robotics; SLAM; Cognition; Mapping; Memory; Attention; Saliency; Focus-of-attention; Object Recognition.

Conteúdo

Capítulo 1

Introdução

Resumo: Este capítulo introduz o tema da tese, assim como os principais problemas que estão associados ao mesmo.

1.1 Enquadramento do trabalho

É do conhecimento geral que desde sempre o ser humano foi criando ferramentas e, mais tarde, máquinas afim de facilitar, melhorar e acelerar a realização das suas tarefas diárias. Da mesma forma que o ser humano foi evoluindo, também as suas criações foram evoluindo, tendo-se chegado ao ponto em que estamos hoje, em que muitas das tarefas mais árduas, demoradas e complexas são realizadas por robôs. No entanto, os robôs desenvolvidos nas últimas décadas, na sua grande maioria seriam apenas destinados à indústria, realizando tarefas sob o controlo e supervisão por parte do homem. Muitas vezes se utiliza o termo "robô autónomo" mas na verdade não existem ainda robôs completamente autónomos. São apenas autónomos no sentido de cumprirem sozinhos determinadas instruções previamente dadas pelo ser humano, não sendo capazes de se adaptar sozinhos a alterações no meio ambiente que os rodeia.

Têm sido muitos os estudos e avanços na área da inteligência artificial no sentido de aumentar a componente autónoma dos robôs de modo a que estes consigam aprender por si mesmos através da interacção com o meio ambiente que os rodeia. Muitos dos avanços feitos nessa área permitem obter bons resultados na resolução de alguns problemas restritos,

mas possuem ainda enormes lacunas quando a quantidade de factores intervenientes nesses mesmos problemas aumenta. No entanto, conhece-se na natureza um sistema capaz de aprender e resolver inúmeros problemas de elevada complexidade: o cérebro humano. Este sistema é capaz de processar de forma rápida e eficiente os diversos estímulos recebidos do meio envolvente, tais como os sons captados pelo aparelho auditivo ou as imagens captadas pelo sistema visual humano, de entre outros. Por outro lado, há que ter em conta que a criação de um modelo computacional que simule o cérebro humano em todas as suas componentes e com todas as suas capacidades é algo extremamente complexo e que certamente ainda levará muitos anos, pois o funcionamento do cérebro humano na sua totalidade é ainda desconhecido.

Uma vez que se considera o cérebro humano como um modelo a seguir quando se pretende desenvolver robôs mais autónomos, é importante focar uma característica que lhe é inerente: a cognição. A palavra cognição geralmente refere-se ao conjunto de actividades mentais dos seres humanos quando lidam com abstracções de informação do mundo real, com as suas representações e armazenamento em memória, bem como com recordações automáticas das mesmas [Patnaik, 2007]. Acredita-se que através da cognição se possa ultrapassar as limitações da inteligência artificial clássica, permitindo-nos criar robôs cognitivos, sistemas robóticos que em vez de serem pré-programados para executar uma tarefa, consigam realizá-la com base nas experiências passadas e nas emoções [Ratanaswasd et al., 2005].

A cognição consiste numa aprendizagem e qualquer aprendizagem começa com uma aquisição de informação. No caso dos seres humanos, essa aquisição de informação começa nos sentidos. Considerando os cinco sentidos humanos, podemos considerar a visão como um dos mais importantes no processo cognitivo e não só, pois são bem conhecidas as grandes dificuldades que as pessoas com deficiências visuais têm no seu dia-a-dia. Apenas com base no sistema visual podemos identificar a grande maioria dos objectos sem tocar neles, sem ouvir o barulho que fazem, sem sentir o seu sabor e sem sentir o seu cheiro. Para além de identificarmos objectos podemos também identificar locais, pessoas e outros seres vivos de uma forma imediata. A visão permite-nos ainda obter noções de distância e tamanho. Outro factor que leva a considerar a visão como um dos sentidos mais importantes é o facto de estar bastante ligada à memória humana. É-nos muito mais fácil reconhecer uma pessoa pela sua cara do que pela sua voz, ou é-nos mais fácil reconhecer um objecto pela visão que temos dele do que pelo que sentimos ao tocar nele.

Fazendo esta análise no sentido inverso, quando pensamos no nome de uma pessoa conhecida, facilmente recordamos a sua face. No entanto, apesar de nos recordarmos facilmente de algumas coisas e podermos rever mentalmente determinados objectos, locais ou pessoas, não nos conseguimos recordar de tudo até ao mais ínfimo pormenor. Algumas coisas ficam registadas na nossa memória de tal forma que nos conseguimos lembrar delas ao longo de toda a vida, enquanto que outras que podem ter acontecido instantes antes e são esquecidas pouco tempo depois. Esta dualidade no armazenamento de informação no nosso cérebro tem sido desde há bastante tempo estudada por investigadores e cientistas ligados à psicologia cognitiva que consideram a existência de dois principais tipos de memória: memória a curto prazo e memória a longo prazo [Patnaik, 2007]. Alguns consideram ainda a existência de um terceiro tipo de memória: a memória sensorial, que consiste na percepção imediata do meio envolvente [Brady et al., 2008].

Um exemplo muito simples da distinção entre os diversos tipos de memória consiste na observação de uma imagem. No momento imediato em que a observámos somos capazes de a descrever até ao mais ínfimo pormenor, pois toda a informação recolhida está bem presente na memória sensorial. Alguns segundos depois apenas conseguimos descrever alguns pormenores, pois só os mais importantes passaram para a memória a curto prazo. Por fim, se formos descrever a imagem dias depois, apenas se conseguirá fazer uma breve descrição da cena, com muito pouco detalhe quando comparada com as memórias sensorial e a curto prazo. Pode dizer-se que, considerando este modelo de memória tripartida, vai havendo uma filtragem de informação recolhida de tal forma que apenas o mais importante fica na nossa memória. Contudo, não se pode afirmar que a memória visual a longo prazo não possa conter um elevado grau de pormenor, pois para além de conseguirmos ter recordações de determinados cenários conseguimos também recordar alguns detalhes que nos permitem fazer uma distinção entre objectos do mesmo tipo e estados dos mesmos.

O nível de detalhe com que o ser humano guarda as suas percepções visuais na memória está dependente da atenção e concentração com que observa o mundo. Enquanto que no dia a dia a maioria dos pormenores acaba por ser desprezada, em situações em que é necessário armazenar um maior nível de detalhe, o ser humano é capaz de o fazer [Brady et al., 2008].

Da mesma forma que o ser humano é naturalmente capaz de seleccionar e armazenar apenas as informações relevantes provenientes das suas percepções, é também capaz de dirigir a sua atenção para o que é relevante, desprezando o resto. Esta capacidade de filtrar a

informação importante é algo que traduz a eficiência do nosso cérebro, pois se o nosso cérebro analisasse toda a informação que os nossos sentidos adquirem estaria constantemente ocupado [Rensink, 2000]. Por exemplo, quando observamos um cenário facilmente identificamos uma série de objectos sem analisar tudo o que se encontra no nosso campo de visão. O mesmo não acontece quando se analisam imagens utilizando métodos convencionais de visão computacional. Nestes as imagens são processadas pixel a pixel, o que torna o processamento extremamente pesado e demorado, quando comparado com as capacidades humanas.

Outro problema que surge quando se tenta modelar a cognição humana no que diz respeito ao sistema visual surge no reconhecimento de objectos. O reconhecimento de objectos é um tema já muito estudado e, tal como acontece em muitos dos problemas descritos aqui, não existe ainda um método que permita efectuar o reconhecimento de objectos no geral. Os métodos mais habituais consistem no reconhecimento de objectos previamente catalogados em bibliotecas de imagens denominadas de imagens de treino [Forssén et al., 2008].

A implementação de um sistema cognitivo em robôs, mesmo que rudimentar, pode dar lugar a uma nova geração de robôs que consigam, de certa forma, adaptar-se a mudanças no meio envolvente e interagir com os humanos. Através de um sistema cognitivo implementado a interacção homem-máquina pode tornar-se muito mais dinâmica e acompanhada da evolução do próprio sistema.

São inúmeras as aplicações que podem surgir para robôs com estas capacidades, tais como a criação de robôs de assistência a idosos ou pessoas com deficiências. De tarefas aparentemente simples como a navegação em ambientes variáveis podem ainda surgir outras aplicações tais como robôs-guia para cegos, em aeroportos ou centros comerciais, robôs de vigilância, de entre outros. São muitas as aplicações mas a cognição na robótica ainda está nos seus primeiros passos. Contudo, a crescente troca de informações entre investigadores, cientistas, engenheiros, psicólogos e neuropsicólogos, de entre outros, certamente culminará na criação de modelos biológicos que se aproximem cada vez mais do modelo cognitivo que sabemos que tem o maior desempenho, mas ainda não conhecemos na totalidade: o nosso cérebro.

1.1.1 Objectivos, contribuições e enquadramento

Neste trabalho pretende-se a criação de um algoritmo de navegação baseado em visão para um robô móvel incluíndo a implementação de um sistema completo de visão, atenção e

reconhecimento de forma a modelar o sistema visual humano e a sua ligação à memória, tendo como objectivo criar uma estrutura que permita a implementação de um modelo inicial de cognição para um robô.

Tenciona-se desta forma criar um modelo com grandes diferenças em relação aos modelos tradicionalmente propostos para resolução de problemas clássicos da robótica móvel tais como o SLAM (*Simultaneous Localisation and Mapping*). Concretamente, pretende-se que o robô seja capaz de navegar num ambiente desconhecido, sendo capaz de o mapear, guardando na sua memória apenas as características relevantes e sendo capaz de reconhecer determinados objectos. Como objectivo final pretende-se que, sendo-lhe dada uma tarefa principal, o robô seja capaz de a realizar sozinho através da conjugação de tarefas já conhecidas.

Como contribuições podem-se salientar:

- Implementação de um sistema de memória composto por memórias a curto e a longo prazo;
- Utilização de um sistema de mapeamento dinâmico, permitindo a actualização do mapa do ambiente através de sucessivos reforços positivos ou negativos;
- Adição de mapas de saliência ao sistema visual do robô, de forma a fazer uma pré-selecção da informação a processar;
- Construção de micro-tarefas e agregação das mesmas de forma a permitir a realização de tarefas mais complexas.

O presente trabalho foi realizado no Vision Laboratory da Universidade do Algarve.

1.1.2 Vista geral

No presente capítulo foi introduzido o tema e foram apresentados os objectivos, contribuições e enquadramento da dissertação. No capítulo 2 foi apresentado o estado da arte e foram introduzidos os conceitos necessários à navegação no ambiente envolvente e o seu mapeamento incluindo os modelos de memória, atenção visual e reconhecimento de objectos. No capítulo 3 foi apresentado o modelo de robô cognitivo, incluindo a gestão de tarefas e os métodos de mapeamento simultâneo. No capítulo 4 foram apresentados alguns dos testes efectuados e foi feita a análise dos dados resultantes. Por fim, no capítulo 5 foram apresentadas as considerações finais e algumas ideias para trabalho futuro.

Capítulo 2

Conceitos gerais e estado da arte

Resumo: Neste capítulo mostram-se alguns dos avanços no ramo da robótica cognitiva realizados nos últimos anos, focando principalmente o problema da localização e mapeamento simultâneo, a estruturação de memória e a realização de tarefas.

2.1 Introdução

Como foi exposto no Capítulo I - Introdução, a robótica cognitiva está ainda nos seus primeiros passos, pois muitos dos trabalhos no ramo da robótica móvel são focados na realização de tarefas específicas, não implicando a necessidade de uma estrutura que permita a um robô aprender e evoluir na realização das mesmas. Como tal, muitas vezes são utilizados uma grande variedade de sensores (infra-vermelhos, ultra-sons, lasers, sensores odométricos, etc.) que apesar de garantirem a realização das tarefas a que se destinam com grande precisão nada têm a ver com a maneira como nós, humanos, percepcionamos o meio que nos rodeia [Montemerlo et al., 2002].

Apesar de o nosso sistema visual nos permitir ter noções de distância e saber se os objectos que nos rodeiam estão perto ou longe não temos capacidade de saber com precisão os valores dessas distâncias. O mesmo acontece quando nos deslocamos de um local para outro. Conseguimos ter uma noção da distância percorrida, mas sem qualquer precisão.

Esta limitação vem agravar um problema clássico da robótica móvel que consiste na localização e mapeamento simultâneos (SLAM - *Simultaneous Localisation and Mapping*)

[Davison et al., 2007], pois sem usarmos métricas exactas torna-se difícil construir um mapa do ambiente que nos rodeia.

Para resolver este problema temos de proceder como os seres humanos e observar o ambiente que nos rodeia e recorrer ao reconhecimento de determinados objectos existentes no meio de forma a criar referências no espaço para que o robô possa calibrar a sua posição. No entanto, numa tentativa de imitar o sistema visual humano [Hubel, 1995], antes de proceder ao reconhecimento de objectos será conveniente a aplicação de um modelo de atenção visual [Itti et al., 1998] de forma a seleccionar as zonas da imagem que possam conter os objectos mais relevantes e estes serem reconhecidos em primeiro lugar. Deste modo o reconhecimento de objectos apenas terá de ser feito nessas zonas da imagem.

Outro dos problemas referido no Capítulo anterior refere-se às mudanças no ambiente e a capacidade que temos de nos adaptar às mesmas devido à estrutura e modo de funcionamento da nossa memória [Deco and Rolls, 2004; Brady et al., 2008]. Apesar de termos uma grande capacidade de armazenamento de informação no nosso cérebro, a informação de que nos lembramos é apenas o essencial do ambiente ou dos objectos. Na robótica clássica os dispositivos electrónicos de armazenamento de dados fornecem uma capacidade de armazenamento mais do que suficiente para a aquisição de toda informação necessária para a realização de tarefas específicas e os sistemas tradicionais de SLAM usam toda essa informação [Montemerlo et al., 2002].

Todavia, quando se considera o caso de um robô que seja capaz de se adaptar às mudanças do meio que o rodeia e aprender sobre o mesmo, o armazenamento contínuo de informação torna-se inviável, pois muita dessa informação poderá ser redundante ou deixar de ter validade após algum tempo, como no caso de uma sala em que a mobília mudou de localização.

Além dos problemas já referidos, é também necessário reflectir sobre o modo de funcionamento do robô, pois tendo em conta que se pretende desenvolver uma estrutura que permita dar lugar à cognição, é necessário que existam vários níveis de acção, de modo a que se possam construir acções complexas a partir da agregação de acções mais simples que podem ser comuns a várias acções diferentes [Ratanaswasd et al., 2005], havendo uma diferenciação em relação à robótica sem fins cognitivos em que apenas são fornecidos aos robôs ciclos de operação complexos.

De uma forma resumida, destacam-se cinco problemas principais que apesar de distin-

tos encontram-se intrinsecamente ligados: (a) o problema do SLAM; (b) organização da memória; (c) modelos de atenção; (d) reconhecimento de objectos; e (e), diferenciação das acções em vários níveis.

Neste Capítulo vamos apresentar cada um destes conceitos, que são fundamentais para a realização do trabalho que será descrito ao longo deste documento.

2.2 Localização e mapeamento simultâneos

A habilidade de simultaneamente localizar o robô (SLAM) e mapear o meio envolvente é um pré-requisito fundamental para o desenvolvimento de robôs verdadeiramente autónomos [Se et al., 2001] [Montemerlo et al., 2002]. Para que o robô possa navegar em ambientes desconhecidos, este tem que ser capaz de ter uma percepção do mesmo que seja suficiente para que possa circular em segurança.

No caso específico desta dissertação a base dessa percepção serão as imagens provenientes da câmara do robô. Imagens essas que serão analisadas afim de se encontrarem objectos ou pontos de referência que sejam importantes a ponto de serem colocados em memória. Contudo, a colocação desses objectos e pontos de referência em memória é algo inútil para a navegação do robô se não tiverem uma localização associada. Deste modo torna-se necessário que o robô vá construindo na sua memória um mapa, de modo a que possa interagir com o mundo à sua volta. O mapeamento do ambiente é também importante para que o robô consiga saber a sua própria localização em relação a um determinado conjunto de referências.

O mapeamento de ambientes para aplicações na área da robótica é um tema já muito estudado, existindo já diversos modelos de mapas aplicados em robôs móveis [Thrun et al., 2000; Davison et al., 2007]. Esses modelos diferem uns dos outros em vários parâmetros, tais como a quantidade de dimensões ou a quantidade e o tipo de referências usadas. Há que ter ainda em conta que muitos dos modelos desenvolvidos têm em conta a aplicação específica que se pretende implementar, não sendo, por isso, adequados a outras aplicações. Os tipos de mapeamento podem ser classificados de acordo com várias características mas existem duas características principais nessa classificação: (a) número de dimensões, pois um mapa pode ser feito a duas ou a três dimensões [Dellaert and Stroupe, 2007; Davison et al., 2007], tendo cada um as suas vantagens e desvantagens. Os mapas a duas dimensões são bastante mais simples, mas por outro lado não permitem efectuar um registo correcto

de determinados pontos de referência no mapa, pois podem haver dois pontos de referência no mesmo local, mas a alturas diferentes [Se et al., 2001]. Por outro lado, os mapas a três dimensões permitem colmatar essa falha mas têm o problema de serem bastante mais complexos. Outra característica que também é bastante utilizada para classificar mapas é (b) o tipo de referências usadas. Um mapa pode ser métrico, topológico [Thrun et al., 1998; Tomatis and Nourkbakhsh, 2001] ou nalguns casos pode até ser híbrido [Buschka and Saffioti, 1998]. Nos mapas métricos faz-se uma representação do ambiente com base num conjunto de coordenadas localizadas num referencial [Thrun et al., 1998]. Este tipo de mapeamento é normalmente muito utilizado quando o robô é dotado de um sistema de odometria e diversos sensores que permitam medir distâncias. Um exemplo deste tipo de mapeamento é o FastSLAM, apresentado por Montemerlo et al. [2002], em que o mapa é criado por um robô dotado de um sistema de odometria e sensores laser, registando em cada ponto do espaço percorrido as distâncias aos obstáculos encontrados. No entanto, erros no sistema de odometria e nos sensores acabam por fazer com que o mapeamento e a localização se tornem cada vez menos precisos ao longo da navegação do robô, pelo que por vezes se torna necessário recorrer à implementação de métodos probabilísticos para reduzir a influência dos erros referidos. Por outro lado, nos mapas topológicos segue-se uma representação baseada na conexão entre determinados pontos de referência, não necessitando assim da aquisição de medidas métricas precisas [Tomatis and Nourkbakhsh, 2001]. Como tal, não pode também oferecer uma localização exacta quer do próprio robô quer dos pontos de referência identificados.

Até agora foram apenas referidos os tipos básicos de mapeamento. Contudo, sobre os mesmos foram desenvolvidas inúmeras variações. Uma dessas variações é proposta por Se et al. [2001] e consiste na utilização de visão *stereo* para detectar um conjunto de pontos chave invariantes a translacções, mudanças de escala e rotações das imagens. Esses pontos são adquiridos utilizando a técnica SIFT (*Scale Invariant Feature Transform*) [Lowe, 1999].

Uma vez que estes pontos são invariantes a uma série de transformações, são boas referências para um robô móvel, pois podem ser detectadas durante a navegação sendo independentes do ângulo de visão. Utilizando as disparidades horizontal e vertical (sistema com 3 câmaras) conseguem saber a localização dos pontos-chave num espaço tridimensional, efectuando o seu mapeamento [Saeedi et al., 2003]. A odometria do robô é também usada no sentido de se saber como é que o robô se moveu, de modo a colocar os pontos de referência no

mapa de forma adequada durante a fase de aquisição. Neste tipo de mapeamento mais uma vez torna-se necessário guardar uma grande quantidade de informação de forma permanente, sendo dada a mesma importância a toda a informação.

Uma outra variante, proposta por Tomatis and Nourkbakhsh [2001], consiste numa arquitectura híbrida (ver Fig. 2.1) que combina os mapas topológicos e métricos num sistema de localização e construção de mapas, criando assim um modelo compacto e que não requer uma elevada consistência métrica a nível global. O modelo do ambiente é caracterizado por dois níveis diferentes de abstracção: os locais são definidos como mapas métricos que permitem a navegação no próprio local; para ir de um local para outro o robô move-se com base no mapa topológico, voltando ao sistema métrico quando chega ao seu destino, necessitando apenas da existência de um ponto de referência métrico para que o robô volte a encontrar a sua posição. Nos modelos topológicos são utilizados pontos de referência tais como os cantos ou aberturas nos caminhos, podendo ser vistos como um conjunto de nós interligados que fornecem a informação necessária para chegar ao destino.

A Figura 2.1 ilustra o mapa topológico utilizado na arquitectura híbrida de Tomatis and Nourkbakhsh [2001]. Este mapa é composto por um conjunto de nós (aberturas) ligados uns aos outros, estando a lista de pontos de referência (cantos) existente entre eles situada no meio dos mesmos. As aberturas (nós) podem ser transições para uma outra divisão ou uma ligação para outro corredor. A cor dos cantos ajuda a distinguir entre cantos com diferentes orientações.

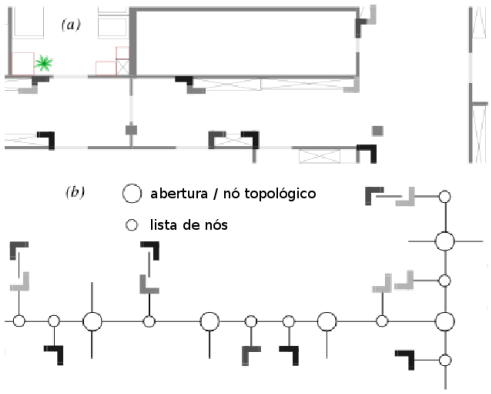

Figura 2.1: Exemplo do mapa topológico utilizado na arquitectura híbrida: (a) Mapa com os cantos e aberturas destacadas. (b) Esquema representativo do mapa topológico. Adaptado da Fig. 2 em [Tomatis and Nourkbakhsh, 2001].

Os pontos de referência são colocados nas ligações entre os diversos nós, permitindo ao robô segui-los para chegar ao seu destino. Esta arquitectura tem bastantes semelhanças com o nosso funcionamento do dia a dia: se um ser humano está numa sala, tem noções das suas distâncias aproximadas aos limites da sala e aos objectos que estejam no seu interior; por outro lado se alguém nos pede informações sobre como chegar a um determinado local, facilmente somos capazes de indicar uma sequência de pontos de referência e direcções a seguir [Kawamura et al., 2002], ou até mesmo de desenhar num papel um mapa facilmente compreensível, mesmo que as distâncias não estejam à escala (ver Fig. 2.2).

Os mapas topológicos enquadram-se também no estudo efectuado por [Vasudevan et al., 2006], que confirmou a existência de uma sequência hierárquica de elementos estruturais quando as pessoas descrevem como ir de um local a outro. Além disso verificou também a utilização de pontos de referência tais como portas e paredes que marcam os limites de um local ou de uma sala.

Outro conceito híbrido apresentado por [Kawamura et al., 2002] visa uma navegação egocêntrica. Este conceito possui ainda uma componente métrica bastante diferente das dos outros métodos. Neste caso o robô no início da navegação deve ter já um esboço do mapa do ambiente na sua memória, com os pontos de referência existentes devidamente assinalados nos respectivos locais. No entanto, o mapa inicial não precisa de ter grande precisão nas medidas, pois a navegação do robô é feita de um ponto de vista topológico. Para além do mapa, é colocada também na memória do robô uma sequência de pontos de referência pelos quais ele tem que passar até chegar ao seu destino.

Um ponto interessante desta proposta é o conceito de egosfera sensorial, que consiste na informação útil do ambiente próximo do robô num determinado momento. Este conceito aproxima-se bastante do conceito de memória a curto prazo, que será analisado mais adiante.

A Figura 2.2 ilustra um esboço de um mapa sem qualquer precisão métrica, mas que permite ao robô navegar graças aos pontos de referência localizados. Está também esboçado o trajecto feito pelo robô na navegação do ponto A ao ponto B [Kawamura et al., 2002].

Existem também propostas como a de Zetzsche et al. [2008] que sugere uma combinação entre as "percepções" do robô e acções motoras, criando-se assim uma arquitectura híbrida, bastante concordante com o comportamento humano, uma vez que os nossos movimentos estão directamente relacionados com a nossa visão. Segundo este modelo, quando o robô se encontra numa fase de exploração do ambiente, em vez de navegar ao acaso, segue o

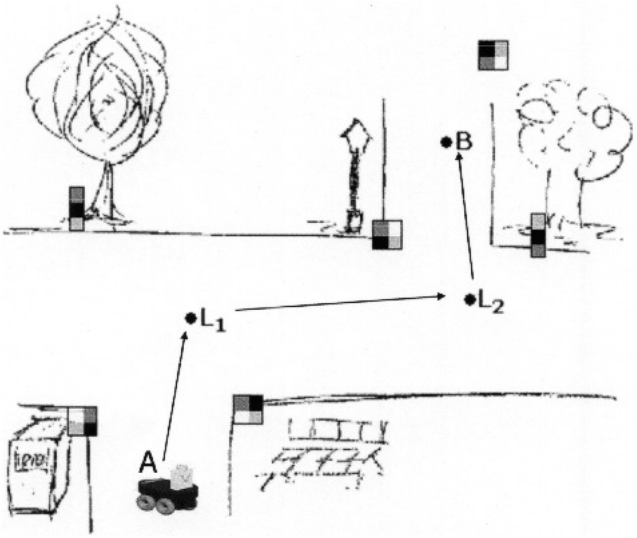

Figura 2.2: Esboço de um mapa sem grande precisão métrica apresentando os diversos pontos de referência e ilustrando o percurso do robô. Adaptado da Fig. 1 em [Kawamura et al., 2002].

caminho no qual encontra maior quantidade de informação, que é de certa forma aquilo que nós, humanos, fazemos. Para tal, estes autores juntam ao sistema de mapeamento um modelo de atenção visual que tem grandes semelhanças com os movimentos sacádicos dos nossos olhos, de forma a encontrar as regiões de interesse.

Como referido no início desta secção, existe uma grande variedade de métodos para a resolução do problema da localização e mapeamento simultâneos. Foram aqui brevemente apresentados os que possuem algumas características que têm alguma semelhança com o comportamento humano, sendo por isso a base mais adequada para implementação de um método apropriado aos objectivos deste trabalho.

2.3 Memória cognitiva

Ao longo dos tempos foram desenvolvidos vários modelos de cognição. De uma forma geral, um modelo de cognição é composto por um conjunto de estados mentais, pelas suas transições e também pela memória cognitiva, sendo esta uma componente fundamental no processo de cognição, pois é inerente aos diversos ciclos nos modelos de cognição. Um exemplo de um modelo cognitivo é proposto por Patnaik [2007], contendo para além da memória sete estados mentais (ver Fig. 2.3): sensoriamento e aquisição, raciocínio, atenção, reconhecimento,

aprendizagem, e, por fim, de planeamento, acção e coordenação.

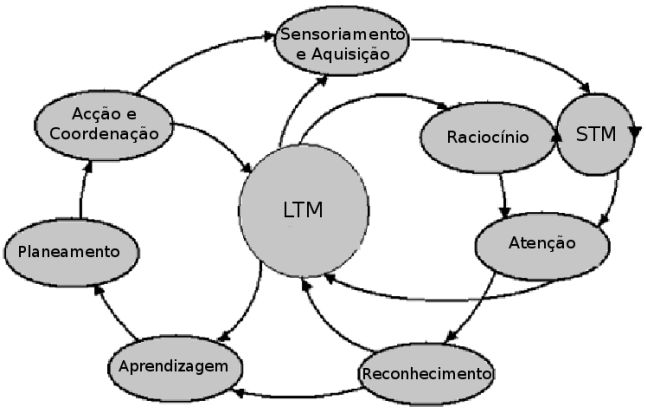

Figura 2.3: Exemplo de um modelo cognitivo em que é possível verificar a existência de três ciclos: aquisição, percepção e aprendizagem e coordenação. O sistema encontra-se centrado na memória a longo prazo (LTM) e está também relacionado com a memória a curto prazo (STM). Adaptado da Fig. 1.1 em [Patnaik, 2007].

É evidente neste modelo o papel central da memória cognitiva, tendo maior destaque a memória a longo prazo (LTM). Existem no modelo três ciclos, sendo a memória a longo prazo um elo de ligação entre os mesmos. O ciclo de aquisição é constituído pelas memórias a curto e longo prazo, pelos estado de sensoriamento e aquisição e pelo estado de atenção [Patnaik, 2007]. O ciclo de percepção consiste nos estados de raciocínio, atenção e reconhecimento e na memória a longo prazo.

Por fim, o terceiro ciclo é composto pelos estados de aprendizagem, planeamento e acção e coordenação e também pela memória a longo prazo. O processo de cognição inicia-se no ciclo de aquisição, em que o ser humano adquire informação através dos seus sentidos, colocando-a na memória a curto prazo para que as componentes com informação mais relevante sejam processadas e enviadas, ou não, para a memória a longo prazo [Patnaik, 2007]. De seguida, no ciclo de aprendizagem o cérebro processa informação previamente adquirida, efectuando o reconhecimento da informação e criando informação de mais alto nível, que é também guardada na memória a longo prazo. Por fim, o ciclo de aprendizagem e coordenação consiste na utilização da informação existente na memória a longo prazo para dar lugar à aprendizagem, à acção e à coordenação [Patnaik, 2007].

Tendo em conta esta importância evidente da memória no processo cognitivo, torna-se

pertinente fazer um estudo da mesma para compreender de que maneira se poderá implementar uma estrutura semelhante num robô móvel.

2.3.1 Tipos de memória

O cérebro humano tem uma enorme capacidade de armazenamento de informação [Brady et al., 2008]. Contudo, algumas dessas informações são preservada por pouco tempo enquanto outras são apenas retidas por períodos de tempo bastante longos. Esta dualidade evidencia a existência de mais do que um único tipo de memória no cérebro humano, pelo que foram feitos vários estudos no sentido de os identificar.

Pode dizer-se que a mente humana possui três tipos diferentes de memória [Brady et al., 2008; Smith et al., 2008], diferenciados pela quantidade de tempo que a informação pode permanecer em memória e também pela origem dessa informação. A memória sensorial serve apenas para armazenar informação sensorial vinda dos sentidos, desvanecendo-se rapidamente a menos que essa informação esteja a ser focada pelo sistema de atenção humano. A memória a curto prazo (*STM - Short-term Memory*) consiste na memória de trabalho do nosso cérebro, ou seja a informação que está directamente a ser processada na nossa mente. Por fim, a memória a longo prazo (*LTM - Long-term Memory*) é onde são armazenadas todas as informações persistentes, permanecendo estas na memória durante longos períodos de tempo mesmo que não sejam utilizadas regularmente [Smith et al., 2008]. A capacidade da memória a curto prazo é bastante pequena tendo em conta a complexidade de algumas tarefas realizadas pelos humanos. Pesquisas recentes têm vindo a indicar que a informação existente na memória de curto prazo consiste num conjunto de referências para informação contida na memória de longo prazo [Smith et al., 2008].

2.3.2 Memória cognitiva na robótica

A utilização de múltiplos tipos de memória (sensorial, STM e LTM) em robótica não é muito comum, pois, uma vez que os computadores possuem uma grande capacidade de armazenamento, muitas vezes não existe preocupação em fazer a gestão da memória de forma a guardar apenas a informação necessária. Nas aplicações habituais faz-se apenas uma acumulação de informação, seja ela muito ou pouco importante para a realização das tarefas para as quais foi programado. Um exemplo disso é evidente na proposta de [Lowe,

1999] referida na secção anterior, em que é necessário acumular uma grande quantidade de pontos de referência para assegurar a navegação de um robô num ambiente desconhecido. No entanto há também outras abordagens em que se tenta modelar o funcionamento da memória humana.

Um exemplo da implementação de uma estrutura de gestão de memória é proposta por Kawamura et al. [2002]. Nesta proposta considera-se a existência de dois tipos de estruturas de armazenamento semelhantes: egosfera sensorial do próprio robô e egosfera sensorial dos pontos de referência pelos quais o robô se deve guiar. A primeira corresponde à memória a curto prazo (STM) e a segunda corresponde à memória a longo prazo (LTM). Deste modo, a STM do robô vai variando enquanto este navega pelo ambiente, enquanto que as egosferas associadas a cada um dos locais de referência pelos quais o robô passe são armazenadas, para consultas futuras. Este armazenamento é feito de forma espacial, de modo a que se crie um mapa com a posição relativa dos diversos pontos de referência conhecidos. A estrutura de memória bipartida utilizada é bastante simples mas é principalmente direccionada para o problema da navegação e localização, contendo lacunas no que diz respeito ao processo cognitivo.

Uma outra abordagem mais complexa no que diz respeito à memória cognitiva é feita por [Ratanaswasd et al., 2005], sendo usadas estruturas de memória de forma a manter a informação necessária para tarefas imediatas e também de forma a armazenar experiências que possam ser usadas durante um processo de tomada de decisão. São usadas duas estruturas principais de memória: STM e LTM.

A STM consiste no armazenamento de informação sensorial numa estrutura denominada de egosfera sensorial. A quantidade de informação armazenada na egosfera sensorial é muito pequena, contendo apenas informação simples como palavras-chave ou cores. Outra característica da STM é o facto de a informação armazenada se ir desvanecendo com o tempo, o que também acontece com o ser humano. Por sua vez, a LTM armazena informação que possa vir a ser usada no futuro. Neste caso considera-se ainda que a LTM está dividida em três partes: memória processual, que armazena primitivas de movimento e comportamento; memória semântica, que é uma base de dados acerca dos objectos existentes no ambiente envolvente; e a memória episódica, que armazena experiências passadas, tais como objectivos ou sequências de tarefas que tenham sido realizadas anteriormente. O conjunto destas estruturas de memória e o inerente fluxo de informação entre elas é denominado de Sistema

de Memória de Trabalho (WMS - *Working Memory System*).

Tendo em conta os exemplos apresentados, é fácil verificar que a estrutura de memória cognitiva proposta por Ratanaswasd et al. [2005] é bastante mais completa e complexa, estando também virada para o processo cognitivo e consequentes acções e comportamentos, em vez de focar apenas o problema da navegação e localização. Contudo, uma estrutura mais simples como a de Kawamura et al. pode ser suficiente para a resolução de tarefas simples.

2.4 Modelos de atenção visual

A maioria das cenas que observamos são complexas demais para serem percepcionadas na sua totalidade. O cérebro humano, apesar de ser extremamente complexo e de ter grandes capacidades também tem as suas limitações e não seria capaz de processar instantaneamente toda a informação captada pelos sentidos, pelo que as suas elevadas capacidades também se devem à sua capacidade de seleccionar e utilizar apenas a informação necessária captada, e não mais que isso. Deste modo, os humanos têm que seleccionar a essência das cenas que observam (*gist*) e processá-las sequencialmente [Martins et al., 2009].

Este processo de selecção não é único dos humanos, estando presente também nos animais, sendo uma característica fundamental para a sobrevivência dos mesmos, permitindo-lhes detectar rapidamente as suas presas e os seus predadores. No entanto, apesar de este processo de capturar apenas a informação essencial ser apenas uma pequena peça no grande *puzzle* que é o funcionamento do sistema visual humano, em conjunto com outras pequenas peças, permite-nos tirar algumas conclusões acerca do funcionamento do mesmo [Martins et al., 2009]: (1) extracção muito rápida da essência da cena observada, (2) extracção também muito rápida da essência de alguns objectos e elaboração de um esboço de um mapa espacial, em paralelo com (3) a construção de um mapa de saliência para Foco-de-Atenção (FoA - *Focus-of-Attention*) e finalmente (4) análise sequencial de regiões divergentes da cena observada para reconhecimento preciso de objectos, utilizando picos e regiões no mapa de saliência com inibição de retorno, de forma a não fixar a mesma região duas vezes.

Este último passo pode ainda decorrer de forma inconsciente, ou de forma consciente (directamente direccionada) [Martins et al., 2009]. O Foco-de-Atenção [Itti and Koch, 2001] é um processo cognitivo que pode ser descrito como a habilidade de concentrar um ou mais

sentidos num objecto específico ou num som. Este processo não ocorre apenas a nível da visão, sendo um exemplo disso o facto de podermos manter uma conversa com uma outra pessoa mesmo que estejamos a ouvir uma grande variedade de sons. O Foco-de-Atenção está directamente relacionado com os movimentos sacádicos dos olhos, que ocorrem inúmeras vezes por minuto, sem que tenhamos consciência disso. No entanto, a existência de informação sobre o local onde determinado acontecimento irá ter lugar poderá ter influência nesses mesmos movimentos [Masciocchi et al., 2009]. Quando olhamos para uma imagem os nossos olhos vão focando sequencialmente e por ordem decrescente de saliência diversos pontos da imagem nas regiões mais salientes (processo exemplificado na Fig. 2.4), existindo determinadas zonas das imagens com maior concentração de pontos observados do que outras.

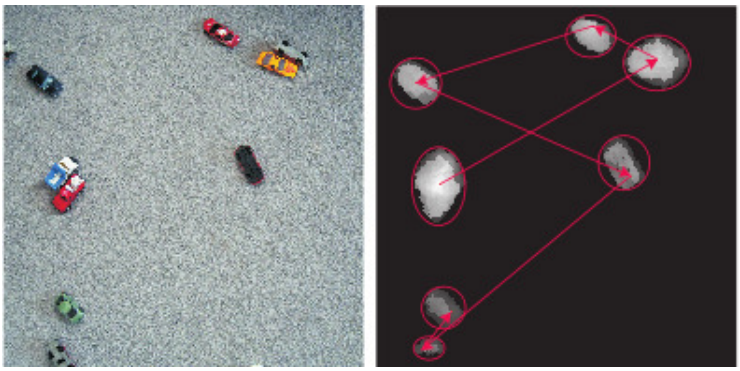

Figura 2.4: Exemplo do processo de Foco-de-Atenção ilustrado no mapa de saliência de uma imagem. A observação das zonas salientes é feita das zonas mais salientes (mais claras) para as zonas menos salientes (mais escuras). Adaptado da Fig. 7. em Martins et al. [2009].

As zonas com maior concentração de pontos geralmente coincidem com regiões de saliência detectadas. Ao conjunto destas regiões quando organizadas topograficamente podemos chamar mapas de saliência. Os mapas de saliência representam a saliência instantânea das diferentes partes de uma cena visual. Estes mapas podem ser construídos com base em várias dimensões, tais como a cor, a orientação ou a intensidade, de entre outras [Itti and Koch, 2001].

Na visão humana, a identificação de regiões salientes na periferia do campo de visão pode ainda dar origem a outros movimentos para além dos movimentos sacádicos dos olhos, tais como movimentos da cabeça. Os movimentos da cabeça permitem direccionar o campo de visão para áreas com maior interesse. Outros factores, tais como o som, podem também influenciar a direcção dos movimentos sacádicos dos olhos e da cabeça. O facto de a atenção

visual guiar o nosso olhar para as áreas com maior interesse faz com que a atenção visual seja um dos mecanismos mais importantes na percepção de um ambiente [Ruesch et al., 2008].

Como já foi referido, os mapas de saliência podem ser construídos a partir de várias características, tais como pontos-chave, contornos, cor, textura, orientação, movimento, disparidade, etc. Contudo, cada uma destas características dá origem a um mapa de saliência diferente, pelo que nalguns casos se combinam os mapas de saliência obtidos a partir de um conjunto de características de forma a obter um mapa que seja a intersecção ou, noutros casos, a reunião das diversas regiões de saliência existentes nos mapas [Martins et al., 2009].

No entanto, nalguns casos consegue-se obter bons resultados utilizando apenas um reduzido número de características. Exemplo disso é o algoritmo para segregação de regiões e saliência proposto por Martins et al. [2008], que se baseia apenas na cor, pois é uma das características das imagens que possui uma maior relevância (ver Fig. 2.5).

Neste algoritmo tenta-se modelar o funcionamento do sistema de atenção humano em situações em que determinados objectos se destacam evidentemente do restante ambiente por possuírem uma cor bastante distinta (não necessariamente homogénea), não se enquadrando na gama de cores predominante na cena observada. O método é feito com apenas 5 passos: (a) normalização da cor, (b) alisamento adaptativo, (c) detecção de contornos, (d) computação do divergência das cores nos pontos de contorno e (e) processamento de geometria de baixo nível. Apesar da sua simplicidade, este método apresentou resultados bastante promissores, tendo sido comparado com uma imagem gerada por 30 observadores humanos, verificando-se que as regiões salientadas pelo algoritmo correspondiam às regiões salientadas pelos observadores.

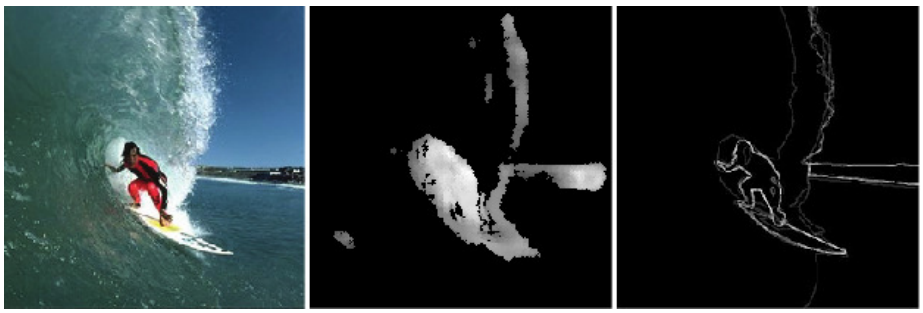

Figura 2.5: Resultados do algoritmo: à esquerda, a imagem original; ao centro, o mapa de saliência obtido; à direita, a imagem gerada pelos 30 observadores. Adaptado da Fig. 1. em [Martins et al., 2008].

Uma característica deste método reside ainda no facto de as regiões ficarem bastante bem definidas, sendo fácil delimitá-las e separá-las do resto da imagem. Na sequência do algoritmo que acabou de ser referido, Martins et al. [2009] apresentou ainda um método biologicamente plausível para obter um mapa de saliência para Foco-de-Atenção, baseado não só em cor, mas também em texturas. O processamento da componente que diz respeito à cor é semelhante ao do algoritmo anterior, combinando-se depois o mapa de saliência resultante com o mapa de textura.

O processamento da textura é, na sua essência, igual ao processamento da cor. Os mapas de saliência obtidos pela cor e pela textura são diferentes mas complementam-se: nos mapas de textura geralmente são realçadas áreas mais difusas, enquanto que nos mapas de cor são mais concentradas nos contornos da imagem. O mapa de saliência final é obtido pela soma pixel a pixel dos valores de saliência da textura e da cor.

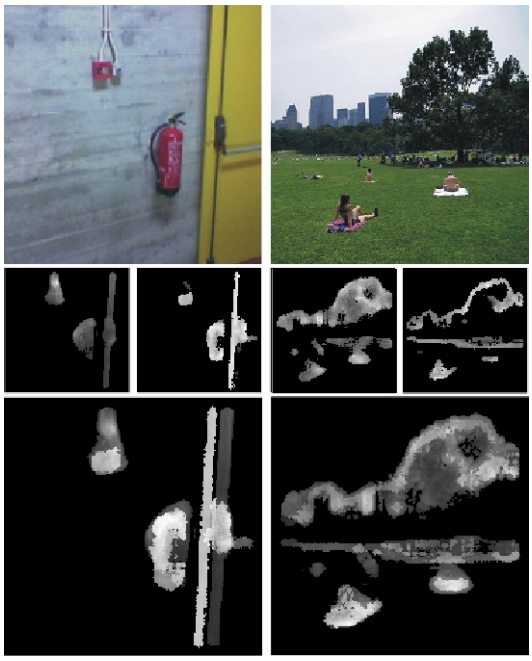

Figura 2.6: Resultados para duas imagens de teste, sendo uma mais simples (esquerda) e outra mais complexa (direita). Na segunda linha, para cada uma das imagens são apresentados os mapas de saliência obtidos a partir da textura (à esquerda) e os mapas obtidos a partir da cor (à direita). Na terceira linha são apresentados os mapas de saliência resultantes. Adaptado da Fig. 6. em [Martins et al., 2009].

Esta combinação da saliência da cor com a saliência da textura faz com que este método seja mais completo que o anterior, pois faz com que sejam destacadas outro tipo de carac-

terísticas nas imagens, tornando assim o método mais robusto a ambientes mais variados.

Na Fig. 2.6 são visíveis os mapas de saliência obtidos através da aplicação deste método a duas imagens. São também visíveis as diferenças entre os mapas de saliência obtidos utilizando a cor e os mapas de saliência obtidos utilizando a textura. No entanto, há que realçar que as imagens apresentadas foram normalizadas para efeitos de visualização, correspondendo as cores mais claras às zonas com maior saliência.

Um algoritmo de atenção bastante diferente é aplicado num robô por Meger et al. [2008]. Este algoritmo faz uso da visão *stereo* para determinar a profundidade do campo de visão de forma a realçar objectos que sobressaiam do chão ou do fundo do campo de visão, criando-se um mapa de saliência com base nas regiões destacadas. Este método implica a utilização de uma câmara *stereo* para gerar os mapas de disparidade. Estas imagens permitem verificar a existência de zonas cuja localização esteja acima do nível do chão.

A Figura 2.7 ilustra o algoritmo de atenção aplicado por Meger et al. [2008], mostrando em cima as imagens captadas pelas duas câmaras, ao meio o mapa de saliência obtido e em baixo a sobreposição do mapa de saliência com a imagem de uma das câmaras.

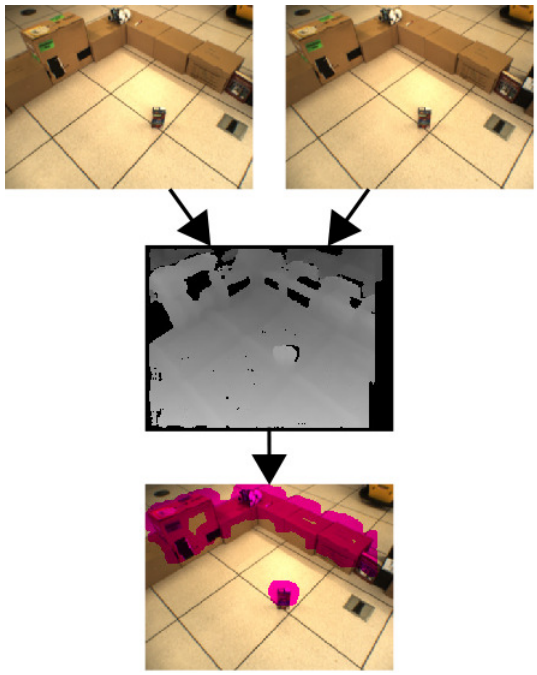

Figura 2.7: De cima para baixo: imagens de entrada da câmara esquerda e da câmara direita, mapa de disparidade, mapa de objectos sobreposto sobre a imagem original. Adaptado da Fig. 2. em [Meger et al., 2008].

Para além do mapa de saliência obtido a partir do mapa de disparidade, é também criado um mapa de saliência baseado na cor, que, em conjunto com o anterior dará origem ao mapa de saliência final. Este mapa final é obtido a partir da intersecção entre os dois mapas.

Um outro algoritmo para criar mapas de saliência foi proposto por Butko et al. [2008]. Este algoritmo consiste numa aproximação a um modelo Bayesiano optimizada para poder ser utilizada em tempo real a baixo custo computacional para direccionar as câmaras de um robô no desenvolvimento de robôs sociais. Para criar o mapa de saliência são usadas as componentes de intensidade da cor, que são depois submetidas a dois filtros (um espacial e um temporal) que apesar de poderem ser implementados sob a forma de uma DoG (*Difference of Gaussians*) foram aproximados para uma DoB (*Difference of Boxes*). A resposta impulsiva dos filtros é depois modelada como uma distribuição Laplaciana com uma variância fixa. Segundo Butko et al. [2008], com o algoritmo *Fast Saliency* o processamento de uma imagem com 320×240 pixéis pode levar apenas cerca de 45 milissegundos com um processador Intel Core Duo 1.87 GHz, que é bastante modesto se compararmos o seu poder de processamento com o dos processadores mais modernos.

De uma forma geral, pode dizer-se que os algoritmos de atenção visual são bastante semelhantes entre si, possuindo bastantes pontos comuns, apesar de poderem partir de um conjunto de características diferentes das imagens ou até mesmo de combinações dessas características. A cor é uma das características mais usadas devido à sua relevância e aos bons resultados obtidos. Contudo, outros aspectos tais como a textura ou a disparidade podem também revelar resultados bastante interessantes e úteis. Verificou-se também que os diversos métodos apresentados seguem uma estrutura semelhante, começando com a criação de um mapa de saliência, sobre o qual será depois aplicado nos tons mais salientes o processo de detecção e reconhecimento de objectos, que será abordado na próxima Secção.

2.5 Reconhecimento de objectos

O reconhecimento de objectos é um tema muito estudado, pois é um tema que está ligado a praticamente todos os ramos da visão computacional. Ao longo dos tempos têm sido feitas pesquisas significantes para desenvolver esquemas de representação e algoritmos com o objectivo de reconhecer objectos em imagens tiradas em condições variadas (ponto de vista, iluminação e oclusão). Em alguns casos de objectos distintos, tais como impressões digitais,

faces e sinais de trânsito, conseguiu-se já obter um sucesso substancial. Um dos objectivos do reconhecimento de objectos está também na categorização dos mesmos.

Na grande maioria das aplicações que envolvem visão artificial é necessário fazer a detecção e reconhecimento de um ou vários objectos. Por exemplo, num ambiente industrial sabe-se que objectos se pretende detectar, e muitas vezes até se sabe em que posição estes vão aparecer, tornando-se a tarefa do reconhecimento bastante mais fácil e directa [Ohali, 2011]. Por outro lado, se os objectos puderem surgir em qualquer posição, a tarefa torna-se mais difícil. Se considerarmos ainda que podem surgir uma grande variedade de objectos, tendo cada um inúmeras vistas possíveis, e com um ambiente de fundo variável, então a tarefa torna-se extremamente difícil se forem utilizados os métodos tradicionais da visão computacional. Outra agravante é ainda a possibilidade de os objectos estarem parcialmente ocultos. Muitos dos métodos de reconhecimento de objectos são baseados na utilização de grandes bibliotecas de imagens, contendo diversas vistas de cada um dos objectos que se pretende detectar [Meger et al., 2008]. Contudo, à medida que se aumenta o número de objectos nas bibliotecas aumenta-se também a ineficiência dos algoritmos de reconhecimento de objectos a nível do tempo de execução, pois aumenta a quantidade de comparações entre as imagens dos objectos existentes nas bibliotecas e a imagem capturada. Outro problema surge nas pequenas variações que os objectos podem ter, mesmo pertencendo ao mesmo tipo de objectos. Tendo em conta todas as complicações referidas, não é difícil concluir que ainda estamos longe de conseguir um sistema de reconhecimento de objectos que se aproxime do nosso sistema visual.

Nós, seres humanos, somos capazes de detectar e reconhecer uma infinidade de objectos quase instantaneamente, independentemente das variações da sua aparência provocadas pela iluminação, posição ou oclusão [Al-Absi and Abdullah, 2009]. Ao olharmos para uma divisão de uma casa imediatamente percebemos se a mesma é uma sala de estar, um quarto ou um escritório, pois reconhecemos imediatamente o conjunto dos objectos presentes nessa divisão, que nos permite tirar conclusões sobre o tipo de divisão que estamos a observar [Vasudevan et al., 2006] ou até fazer a previsão de que outros objectos esperamos encontrar nessa divisão. Além disso, somos ainda capazes de categorizar objectos mesmo que nunca tenhamos visto esses mesmos objectos. Por exemplo, sabemos dizer que uma cadeira é uma cadeira mesmo que nunca tenhamos visto a cadeira em questão.

Na próxima secção far-se-á um estudo do funcionamento do sistema visual humano no

que diz respeito ao reconhecimento de objectos. Após este estudo descrever-se-ão alguns métodos de reconhecimento de objectos já implementados.

2.5.1 Reconhecimento de objectos por humanos

O reconhecimento de objectos é uma habilidade que os seres humanos possuem desde a infância. Com uma simples observação de um objecto os humanos são capazes de fazer a sua identificação e categorização apesar das inúmeras alterações que podem existir no objecto devido à iluminação, posição ou oclusão. No entanto, é um enorme desafio conseguir desenvolver sistemas de visão que consigam equiparar as capacidades cognitivas dos seres humanos, ou sistemas que sejam capazes de identificar um objecto específico que esteja a ser observado. As principais dificuldades no desenvolvimento deste tipo de sistemas devem-se às variações na iluminação, à posição do objecto e na dificuldade em fazer a generalização de um objecto a partir de um conjunto de imagens de exemplo.

A forma como o sistema visual humano efectua o reconhecimento de objectos ainda não é completamente conhecida. Contudo, tem sido alvo de muito estudo, que resultou na elaboração de algumas teorias sobre o seu funcionamento. Alguns dos estudos foram feitos com recorrência a equipamentos dispendiosos [Rodrigues, 2008] de forma a verificar os níveis de actividade em determinadas zonas do cérebro quando determinadas imagens eram apresentadas ao observador.

Por outro lado, outros estudos foram bem mais simples, consistindo na análise de determinados comportamentos humanos em determinadas situações numa tentativa de perceber que factores têm maior importância no reconhecimento de objectos e de que forma são armazenados na nossa memória. Um desses estudos foi realizado por [Vasudevan et al., 2006] e permitiu concluir que a estrutura dos objectos foi considerada o aspecto fundamental dos mesmos para efectuar o seu reconhecimento. Uma estrutura de um objecto pode ser representada por pontos, linhas e contornos, havendo já alguns métodos que utilizam este tipo de representação para efectuar o reconhecimento de objectos, que serão abordados na próxima secção.

A nível do processo de reconhecimento, muitas vezes se pensou no sistema de visão humano como uma sequência de processos (detecção, segregação, categorização e reconhecimento). Contudo, estes não podem ser completamente sequenciais [Bar et al., 2006]. Os vários processos têm que ocorrer em "paralelo", pelo menos parcialmente. Era um hábito

comum pensar-se que para efectuar o reconhecimento de um objecto era necessário isolá-lo primeiro do ambiente de fundo. Contudo, pesquisas recentes sugerem que a categorização dos objectos ocorre antes ou ao mesmo tempo que a segregação dos mesmos, ou seja, na altura em que temos consciência do que é o objecto que está a ser observado, o cérebro já sabe que objecto é [Rodrigues and du Buf, 2009a].

Apesar destes avanços e pesquisas ainda se desconhece a ordem exacta pela qual os processos ocorrem, assim como os casos em que existe paralelização de processos. Apesar da complexidade e das múltiplas incógnitas do sistema visual humano, é certo que o reconhecimento de objectos não pode ser encarado como uma única tarefa simples. Este processo tem que ser encarado como uma tarefa com vários níveis [Rodrigues and du Buf, 2009a], podendo existir em cada nível mais do que um processo a decorrer em simultâneo.

Os avanços que se têm vindo a fazer no conhecimento do sistema visual humano têm servido de inspiração para a criação de vários modelos biológicos que tentam modelar a visão humana. Contudo, estes modelos exigem geralmente muito processamento, sendo necessária a utilização de processadores com elevados desempenhos. Estas exigências a nível de processamento fazem com que os modelos biológicos não tenham sido ainda muito aplicados em aplicações de tempo real, como é o caso da robótica.

2.5.2 Métodos de reconhecimento de objectos

Existem vários métodos de reconhecimento de objectos, podendo-se considerar que existem dois tipos principais: os dedicados e os gerais. Os métodos dedicados são desenvolvidos com o objectivo de reconhecer um número limitado de objectos, sendo por isso optimizados em função dos objectos a detectar. Este tipo de métodos são os que são geralmente utilizados em ambientes industriais para inspecção de produtos e monitorização de processos [Ohali, 2011]. Por sua vez, os métodos mais gerais podem funcionar na grande maioria das aplicações, geralmente com um maior custo computacional [Lowe, 1999; Evans, 2009].

Há que ter em conta que para a grande maioria das aplicações dos dias de hoje, os algoritmos de carácter meramente computacional são mais eficientes, pois são específicos para a realização de uma determinada tarefa. No entanto, quando se começa a caminhar para o desenvolvimento de sistemas cognitivos é inevitável a utilização de métodos mais versáteis, pois permitem uma melhor adaptação a situações mais complexas.

Os métodos de reconhecimentos de objectos têm como base a extracção e reconhecimento

de regularidades das imagens, tiradas sob diferentes iluminações e posições. Por outras palavras, a generalidade dos algoritmos adopta certas representações e modelos para capturar essas características, facilitando assim a execução de procedimentos de identificação dos objectos [Lowe, 2004]. As representações podem ser tanto modelos geométricos a 2D ou a 3D. Após a extracção das características fundamentais da imagem , é feito o processo de reconhecimento, que se baseia na comparação dos modelos ou representações dos objectos com a imagem de teste [du Buf et al., 2010].

As características mais comuns nas quais os algoritmos de reconhecimento de objectos se baseiam são a geometria, o aspecto e os pontos de interesse. No que diz respeito à geometria dos objectos, são extraídas das imagens primitivas geométricas (linhas, círculos, etc.) que sejam invariantes à perspectiva de observação. Por sua vez, os algoritmos baseados no aspecto utilizam como base os padrões existentes nos objectos (características, texturas, histogramas, etc.)[Shotton et al., 2008]. Por fim, os métodos baseados nos pontos de interesse consistem na procura de determinados pontos nas imagens que são invariantes às alterações provocadas por mudanças de escala ou de iluminação. Um dos métodos de representação mais utilizados para aplicações de visão é o SIFT (Scale-Invariant Feature Transform), proposto por Lowe[1999].

A partir desta forma de representação foi desenvolvido um sistema de reconhecimento que transforma a imagem numa grande colecção de vectores de pontos essenciais que são invariantes às mudanças de escala, às translações e às rotações e ainda parcialmente invariantes às mudanças na iluminação e nas projecções 3D. Estes pontos são obtidos a partir de uma série de filtros que resulta na identificação de pontos estáveis. O primeiro passo do algoritmo consiste em: (a) identificar pontos-chave na imagem que sejam extremos de uma função de diferença de Gaussianas, em múltiplas escalas; (b) localizar os pontos-chave nos locais onde foram encontrados os extremos determinando também a sua escala; (c) definir a orientação dos mesmos através dos gradientes locais da imagem; e (d), construir os descritores dos pontos-chave através da medição dos gradientes locais de uma região vizinha a cada ponto de interesse.

O facto de estes pontos serem invariantes a uma grande quantidade de alterações, resultou na criação de um método robusto de reconhecimento de objectos mesmo em casos de oclusão de partes dos mesmos em ambientes desorganizados e com fundos complexos.

Após este primeiro passo, que é a extracção dos pontos SIFT, procede-se à verificação da

correspondência, ou *matching*, entre os pontos SIFT do objecto em análise e os pontos SIFT dos objectos existentes na base de dados de objectos. De seguida, de forma a remover alguns pontos-chave que possam dar origem a erros utiliza-se uma transformada de Hough para agrupar cada correspondência (*clustering*) de todas as imagens da base de dados, dependendo da transformação particular a que esteja submetida (transformação, rotação e mudança de escala). No final, todos os grupos, ou *clusters*, com pelo menos três correspondências para uma imagem em particular são aceites [Ramisa et al., 2008].

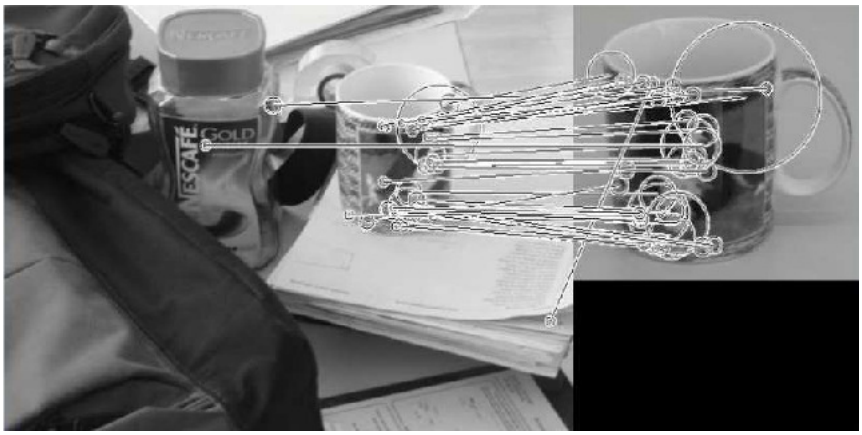

Figura 2.8: Processo de *matching* no método de reconhecimento de objectos referido acima. Adaptado da Fig. 1. em [Ramisa et al., 2008].

O método SIFT, proposto por Lowe [2004] foi testado e analisado por Ramisa et al. [2008], tendo sido comparado com o método *bag of features* proposto por Nistér and Stewénius [2006]. Este método faz uma espécie de analogia com os métodos da categorização de texto, em que a ocorrência de determinadas palavras em documentos é registada e utilizada para treinar classificadores que poderão depois reconhecer o assunto de outros textos. Neste caso em vez de palavras têm-se descritores locais, como os SIFT. Esses descritores locais são então utilizados para gerar um histograma que caracteriza a imagem em análise. De modo a limitar o tamanho dos histogramas, é criado um dicionário de grupos de pontos, ou *clusters*. Por fim, é feito o *matching* entre o histograma da imagem em análise e os histogramas das imagens existentes na base de dados.

Neste estudo houve a particular preocupação com a exigência dos algoritmos a nível de processamento, pois ambos têm em vista a aplicação na robótica móvel, tendo o processamento que ser feito em tempo real no próprio robô e não num servidor externo.

Dos resultados obtidos por Ramisa et al. [2008], verificou-se que para os objectos tex-

turados e uniformemente texturados os resultados são bastante semelhantes para ambos os algoritmos. Contudo, é importante realçar que o método de Lowe [2004] não conseguiu efectuar o reconhecimento de objectos não texturados durante os testes. No entanto, a nível do tempo de processamento o algoritmo de Lowe [2004] foi superior, sendo o processamento bastante mais rápido que no método de Nistér and Stewénius [2006].

Existe ainda um outro tipo de descritores baseados no SIFT, que são os descritores SURF [Bay et al., 2008]. O algoritmo que gera os descritores SURF foi desenvolvido com o objectivo de ser mais leve a nível de processamento, tendo assim um melhor desempenho para aplicações em tempo real. Este algoritmo é composto por três etapas: (a) criação da integral da imagem; (b) determinação de pontos de interesse através de *Fast-Hessian*; e (c) criação do descritor de cada ponto-chave. Segundo o autor, o algoritmo SURF é mais rápido, mais robusto e mais preciso que o algoritmo SIFT.

Um outro método que não serve apenas para reconhecimento de objectos, mas também para a sua categorização, é proposto por Rodrigues and du Buf [2009a]. Todavia, neste caso é biológico (tratando-se de um modelo cortical). Este método é baseado em características multi-escala: linhas, arestas e pontos-chave são extraídos das respostas de células simples, *complex* e *end-stopped* na área cortical V1. Os pontos-chave são utilizados para construir mapas de saliência para posterior aplicação do processo de Foco-de-Atenção.

O método proposto permite obter translações 2D, rotações e invariância no tamanho através do mapeamento dinâmico de mapas de saliência baseados em informação proveniente dos pontos-chave multi-escala. O modelo é funcional e está dividido em duas partes, sendo que os pontos-chave são usados para localizar possíveis objectos enquanto que as linhas e as arestas são utilizados para o processo de reconhecimento e categorização. Além desta divisão em duas partes existe também uma progressão no detalhe de análise dos fluxos de dados, começando-se em ambos os casos numa escala mais grossa (menos detalhe; baixas frequências) e progressivamente ir passando a uma escala mais fina (mais detalhe; altas frequências).

Todos os processos descritos até agora neste método têm uma analogia directa com o funcionamento do córtex cerebral no que diz respeito ao sistema visual humano. A evolução deste modelo cortical poderá resultar na abrangência de um maior número de aspectos cognitivos num futuro próximo, devido à sua grande componente biológica.

Na Figura 2.9 estão visíveis alguns exemplos da detecção de linhas e arestas em multi-

escala. Em cima e à esquerda encontram-se as escalas mais finas e à direita as escalas mais grossas. Ao meio e à esquerda está a imagem inicial de uma caneca e à direita a sua reconstrução através da combinação das componentes de baixas frequeências e interpretação simbólica das linhas e arestas em algumas escalas (terceira e quarta imagens). Em baixo está a representação dos pontos-chave em multi-escala da caneca com escalas mais finas à esquerda e escalas mais grossas à direita.

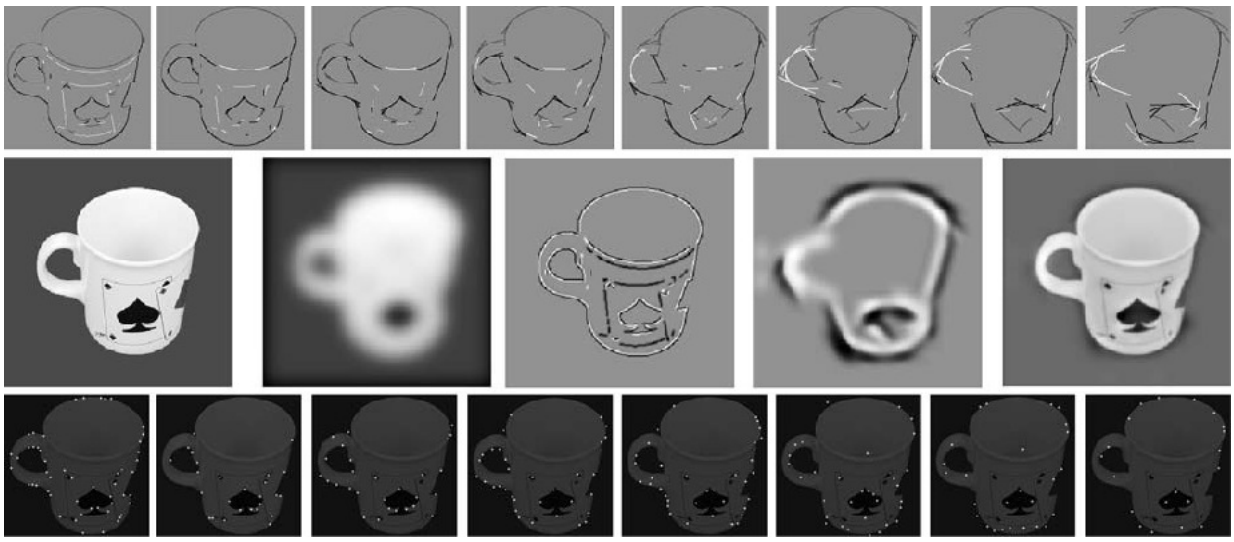

Figura 2.9: Topo: detecção de linhas e arestas em multi-escala para o caso de uma caneca. Adaptado da Fig. 1. em [Rodrigues and du Buf, 2009a].

Tendo em conta os métodos apresentados, pode considerar-se que no geral, são todos aplicáveis para o objectivo deste trabalho. O método cortical de Rodrigues and du Buf [2009a] parece ser o mais biologicamente plausível mas os elevados requisitos a nível de processamento tornam-no inviável para aplicações em tempo real. No entanto, o método de Lowe [2004] e o de Nistér e Stewénius [2006] exigem menos processamento, sendo possível aplicá-los em tempo real na robótica móvel.

2.6 Gestão e realização de tarefas

Uma outra característica da cognição humana é o facto de conseguirmos planear e executar sequências de acções de forma a atingir um objectivo. Mesmo que a sequência de acções necessária não nos seja transmitida, em muitos casos somos capazes de inferir o conjunto de acções a realizar a partir de experiências passadas [Meinert, 2008]. Outra característica

relevante é a capacidade de construir tarefas complexas a partir da agregação de tarefas mais simples, que por sua vez são também construídas a partir de tarefas ainda mais básicas, havendo uma grande variedade de níveis de complexidade. De acordo com estas características torna-se necessário implementar um sistema de gestão de tarefas que permita a existência de múltiplos níveis de complexidade e a agregação de tarefas mais básicas já conhecidas em tarefas mais complexas.

Os esforços no desenvolvimento de sistemas deste género não são ainda muito numerosos e como tal consiste ainda num grande desafio. Na robótica tradicional a execução de procedimentos pré-definidos era suficiente na grande maioria dos casos e noutros, em que era necessária alguma aprendizagem, recorria-se a métodos de inteligência artificial que permitiam a aprendizagem para a realização de uma determinada tarefa mas sempre com base em probabilidades ou cálculos matemáticos que acabam por não ter qualquer semelhança com o comportamento e raciocínio humano. Outro problema destes métodos é não serem apropriados a um ambiente dinâmico, em constante mudança.

De acordo com Ratanaswasd et al. [2005], as acções têm que ser seleccionadas cuidadosamente utilizando informação baseada nas experiências passadas, na tarefa e no ambiente. os autores apresentaram um modelo de controlo cognitivo em que juntou às memórias a curto e longo prazo uma memória procedimental (também com duração de longo prazo) em que seriam armazenados procedimentos de algumas tarefas a executar que depois poderiam ser combinadas para realizar outras tarefas mais complexas. A gestão do planeamento e execução das tarefas seria feita por uma estrutura denominada de Agente Executivo Central. A escolha das tarefas a realizar seria elaborada através de um sistema de recompensas, sendo executadas as tarefas que dessem origem à maior recompensa de acordo com a tarefa exigida.

Um outro sistema de gestão, planeamento, selecção e execução de tarefas é proposto por Alami et al. [2006]. Este sistema tem também uma componente responsável por controlar e monitorizar todo o comportamento do robô. Também neste caso existe uma série de tarefas simples, tais como que são depois encadeadas de forma a realizar outras tarefas.

Uma forma ligeiramente diferente de abordar o problema foi desenvolvida por Jung et al. [2007]. Os autores criaram uma base de dados de conhecimento e regras sob a forma de *scripts*(ver Fig. 2.10). Quanto maior a quantidade de *scripts* armazenada, maior seria a flexibilidade do robô e a variedade de tarefas que o mesmo poderia realizar. A flexibilidade

advém do facto de se poder decompor uma tarefa hierarquicamente em *scripts* básicos que podem ser recombinados de forma a dar origem a uma tarefa diferente. No entanto também existe flexibilidade na inserção de *scripts*, pois torna-se assim mais fácil de aumentar a base de conhecimento do robô. Além disso utilizando este tipo de sistema torna-se possível contextualizar o conhecimento. Todos os *scripts* são compostos por uma pré-condição e uma pós-condição que permitem dar início ou terminar os mesmos. Além disso existe também uma condição relacionada com o ambiente, de forma a que o robô possa seleccionar a informação relacionada com esse ambiente.

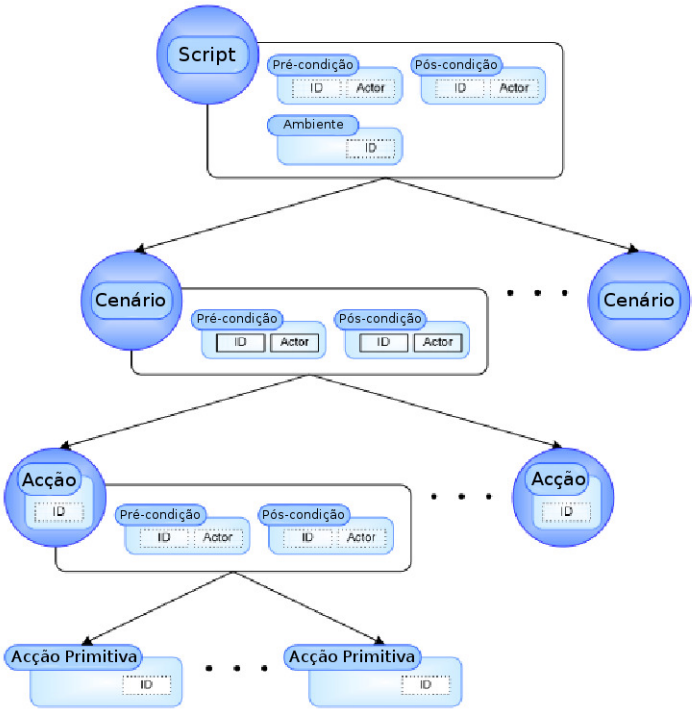

Figura 2.10: Exemplo da estrutura de um *script* de uma tarefa. Adaptado da Fig. 2. em [Jung et al., 2007].

Para além destas três características, os *scripts* têm depois uma sequência de cenários que têm também uma pré e uma pós condição que por sua vez resultam em acções que têm também as suas pré e pós condições. As acções consistem em sequências de acções primitivas. Esta forma de estruturar a realização de tarefas é bastante versátil e permite que o comportamento do robô siga uma espécie de narrativa ou uma espécie lógica de causa-efeito.

A Figura 2.10 representa um *script*, que pode ser representado como uma representação estruturada que descreve a sequência de acções e o ambiente num contexto particular. Um

script é composto por quatro propriedades: pré-condição, pós-condição, ambiente e cenário. As propriedades de pré-condição e pós condição são as condições que iniciam e terminam o *script*, respectivamente. A propriedade de ambiente serve para contextualizar as acções do robô. Por sua vez, cada cenário é composto por uma série de acções e cada acção é associada a uma situação dentro do próprio cenário. Cada acção é ainda composta por um conjunto de acções prmitivas.

Após a apresentação dos conceitos base e dos métodos aplicáveis à robótica cognitiva, no capítulo seguinte será descrita a implementação da arquitectura cognitiva proposta nesta dissertação.

Capítulo 3

Arquitectura do Robô Cognitivo

Resumo: Este capítulo faz uma breve descrição da implementação das metodologias propostas, começando pela caracterização do robô, métodos aplicados para a criação do mapa de saliência e reconhecimento de objectos, continuando depois com os mecanismos de realização e construção de tarefas. Por fim é feita uma descrição detalhada do sistema de localização e mapeamento simultâneo implementado no robô. Neste sistema utiliza-se a informação visual obtida pelo robô combinando o mapa de saliência e o reconhecimento de objectos, por forma a retirar informação que é depois colocada nas estruturas de memória que, por sua vez, são a base do sistema de SLAM proposto.

3.1 Caracterização da plataforma robótica

Para a implementação e teste do sistema cognitivo utilizou-se o robô Surveyor SRV-1 com um sistema de visão *stereo* (ver Fig. 3.1 à esquerda). O robô é de reduzidas dimensões ($25 \times 16 \times 13cm$) e como método de locomoção tem duas lagartas actuadas por dois motores DC cada uma, utilizando assim tracção diferencial.

O sistema de visão *stereo* consiste numa placa com duas câmaras montada sobre uma estrutura com dois motores servo, que permitem fazer o *pan* e o *tilt*, podendo-se assim direccionar as câmaras tanto horizontalmente como verticalmente.

Cada uma das placas das câmaras possui um processador Blackfin de 500 MHz que pode

Figura 3.1: À esquerda o robô SRV-1 com sistema de visão *stereo*, ao centro o robô na posição P1 e à direita na posição P2(ver texto para mais detalhes).

ser usado para interpretar programas escritos em linguagem C ou para correr *firmware* que permite aos utilizadores controlar o robô utilizando um computador remoto ligado por Wi-Fi. As câmaras permitem obter imagens JPEG com resoluções entre 160×128 e 1280×1024 pixéis num máximo de 60 *frames* por segundo (nas resoluções mais baixas).

Para a realização do trabalho descrito nesta dissertação configurou-se o robô para se ligar a uma rede Wi-Fi, podendo-se assim obter as imagens capturadas pelas câmaras do mesmo e controlá-lo utilizando um computador ligado à mesma rede. O *firmware* do robô funciona como um servidor TCP que está constantemente à espera de pedidos de ligação por parte de clientes TCP que, por sua vez, enviam comandos predefinidos que são depois interpretados pelo *firmware* de forma a realizar a acção correspondente ao comando recebido. Todos os comandos enviados para o robô consistem em caracteres ASCII ou caracteres ASCII seguidos de caracteres binários. O robô responde a todos os comandos com mensagens de confirmação.

Por exemplo, quando se pretende actuar os motores do robô é necessário enviar um comando com o formato 'Mabc' em que a e b correspondem às velocidades de rotação dos motores da esquerda e da direita, respectivamente, e c corresponde à duração do movimento, sendo a duração do movimento $c \times 10ms$.

A recepção deste comando é depois confirmada pelo robô com o envio da mensagem '#M'. O *firmware* do robô suporta uma grande variedade de comandos, permitindo assim controlar o movimento do robô, os movimentos dos dois servos que controlam "a cabeça," o

tamanho e qualidade das imagens transmitidas, saber o estado da bateria e até fazer algum processamento nas imagens antes de proceder ao seu envio (detecção de arestas, segmentação por cor, detecção da linha do horizonte, de entre outras). Contudo, nenhum dos comandos de pré-processamento foi utilizado neste trabalho.

De modo a simplificar o estabelecimento de ligações TCP e o envio dos comandos para o robô foi criada uma biblioteca de funções em ANSI C denominada de Camada de Abstracção do Hardware. Esta biblioteca fornece ao programador um interface mais intuitivo, permitindo assim que possam ser enviados comandos para o robô com a simples chamada de uma função cujos argumentos são unidades de fácil manipulação para o utilizador. Utilizando o mesmo exemplo referido acima relativo à actuação dos motores, com esta biblioteca basta apenas chamar a função *HAL_move_to(d,s)* em que d é a distância pretendida, em *cm*, e s é a velocidade pretendida, em *cm/s*, sendo a conversão dos argumentos para os valores enviados para o robô feita de forma transparente para o programador.

A nível da movimentação do robô é necessário salientar que este não tem qualquer sistema de odometria. Apesar de as instruções serem dadas em *cm* o movimento do robô nunca tem precisão, estando sempre dependente do estado da bateria, do atrito entre o chão e as lagartas e da inclinação da cabeça do robô (faz variar o seu centro de massa, afectando o movimento uma vez que o robô é bastante leve). O mesmo acontece nas rotações, em que as instruções são dadas em graus, podendo haver desvios relativamente grandes. A única maneira de controlar a quantidade de movimento é com base no parâmetro de tempo (c) referido acima, permitindo-nos ter uma noção aproximada da quantidade de movimento, mas nunca precisa, pois o robô não possui qualquer sistema que forneça um retorno de informação acerca do seu movimento.

É de salientar neste ponto que apesar de dificultar bastante a navegação e o mapeamento este problema acaba por estar de acordo com os objectivos propostos para esta dissertação, que consistem em criar um sistema baseado no ser humano pois nós próprios não sabemos quantificar com precisão as nossas deslocações ou "rotações," conseguimos apenas ter estimativas.

No que diz respeito ao controlo da "cabeça" tanto a inclinação como a rotação da mesma contêm 5 posições. Contudo, na inclinação da cabeça apenas são usadas as duas posições em que a mesma fica mais inclinada para baixo (posição P1, posição 1 de 5, ver Fig. 3.1 ao centro), de forma a captar as imagens da zona do ambiente mais próxima, e a posição que

corresponde à menor inclinação da cabeça (posição P2, posição 3 de 5, ver Fig. 3.1 à direita), permite observar um pouco acima da linha do horizonte, pelo que permite a visualização do ambiente a distâncias grandes relativamente ao tamanho do robô. Por sua vez, no que diz respeito à rotação da cabeça são usadas as 5 posições (2 para a esquerda, 1 para o centro e 2 para a direita), pois o reduzido ângulo de visão das câmaras (65º) torna necessária a utilização das mesmas frequentemente.

De forma a obter uma boa velocidade de aquisição e processamento as imagens, $I(x, y)$, foram captadas com a resolução de $M \times N$ (320×240), com $x = \{1, ..., M\}$ e $y = \{1, ..., N\}$. Apesar do reduzido tamanho, é suficiente para a grande maioria dos casos, tendo em conta o também reduzido tamanho do robô. Por outro lado, no que diz respeito à qualidade JPEG foi necessário optar pela qualidade máxima, uma vez que as qualidades mais baixas, apesar de tornarem o envio das imagens mais rápido, faziam com que os blocos de compressão JPEG ficassem visíveis na imagem, o que pode provocar o mau funcionamento dos algoritmos de processamento de imagem (detecção de arestas, por exemplo).

Para descodificar as imagens JPEG recebidas, de forma a obter imagens PGM ou PPM, utilizou-se a biblioteca *opensource libjpeg*. As imagens PPM obtidas após a descompressão encontram-se no espaço de cor RGB, podendo-se assim definir cada pixel P_i da imagem I como (R_i, G_i, B_i), sendo $i = \{1, ..., M \times N\}$.

Para a realização do trabalho a que esta tese se refere foi necessária a criação de uma série de bibliotecas de funções em C e C++. Estas bibliotecas apesar de corresponderem a componentes distintas da programação do robô, tais como hardware, manipulação de ficheiros de imagem, algoritmos de processamento de imagem, realização de tarefas ou mapeamento, são, na generalidade, interdependentes.

Para além das bibliotecas criadas, são também utilizadas duas bibliotecas *opensource* escritas em C/C++: OpenSURF [Evans, 2009], que é utilizada para efectuar o reconhecimento de objectos; e NMPT (*Nick's Machine Perception Toolbox*)[Butko, 2008], que é utilizada para a criação de mapas de saliência. A biblioteca OpenSURF consiste numa implementação optimizada em termos de tempo de execução do algoritmo SIFT (ver Secção 2.5.2). Ambas as bibliotecas utilizam componentes da biblioteca OpenCV, que é uma biblioteca de visão computacional (http://opencv.willowgarage.com/).

Na Fig. 3.2 estão representadas as diferentes camadas de *software* consideradas:

(a) Camada de Abstracção do Hardware: consiste num conjunto de funções de baixo nível

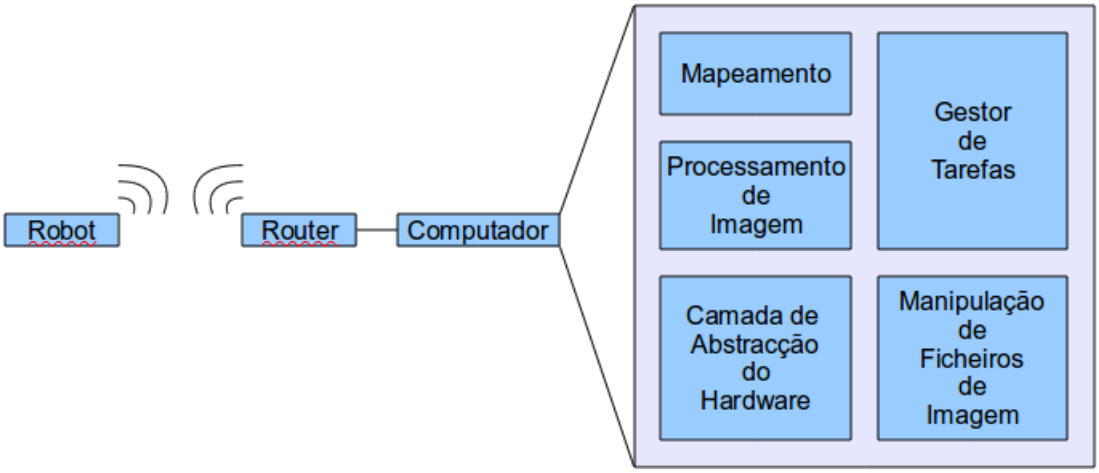

Figura 3.2: Esquema representativo das camadas de software.

responsáveis por efectuar a ligação TCP ao robô, envio de comandos para a realização de acções motoras ou para alterar configurações e recepção de imagens ou outras informações tais como o estado da bateria;

(b) Manipulação de Ficheiros de Imagem: esta camada contém uma série de funções que permitem realizar as operações básicas de ler e gravar ficheiros de imagem nos formatos PGM e PPM, tanto em binário como em ASCII;

(c) Processamento de Imagem: é composta pelas funções de processamento de imagem necessárias para realizar todas as operações de detecção e reconhecimento efectuadas pelo sistema visual do robô;

(d) Gestor de Tarefas: consiste num agente central que faz o agendamento e o planeamento das tarefas que o robô tem de realizar;

(e) Mapeamento: esta componente contém todas as funções necessárias para o robô se poder auto-localizar e mapear o ambiente onde navega.

O robô descrito nesta Secção foi testado num ambiente especificamente preparado para o mesmo, ou *sandbox* (ver Fig. 3.3), pois tendo em conta o seu reduzido tamanho não seria viável testar a arquitectura proposta em ambientes reais tais como corredores ou salas. Espalhados pela *sandbox* foram colocados diversos objectos para serem reconhecidos e utilizados como pontos de referência. A *sandbox* é limitada por fita verde no chão, sendo

considerada como uma parede para o robô. No entanto, a arquitectura apresentada pode ser facilmente adaptada para um ambiente real utilizando, por exemplo, um algoritmo semelhante ao utilizado por José et al. [2010] para a detecção de corredores em vez da detecção da fita verde.

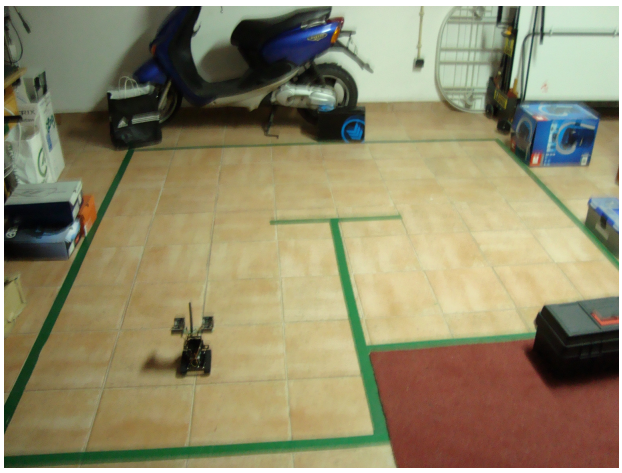

Figura 3.3: Exemplo de um ambiente onde o robô se movimenta (*sandbox*).

Nesta Secção fez-se a caracterização da plataforma robótica tanto a nível de *hardware* como de *software*. Nas Secções seguintes irá ser feita a descrição da arquitectura cognitiva proposta neste trabalho, que pode ser resumida pelos seguintes passos: (a) detecção da saliência e procura do objecto mais saliente; (b) reconhecimento dos objectos pelo SURF; (c) gestão de tarefas; terminando na (d) localização e mapeamento simultâneo.

3.2 Saliência

Com o objectivo de seleccionar as regiões das imagens que possam conter informação útil foi necessário aplicar um algoritmo de saliência, permitindo assim a pré-selecção de regiões de interesse nas imagens captadas. Cada uma dessas regiões pode depois ser processada para detecção de obstáculos e reconhecimento de objectos, em vez de ser necessário processar as imagens completas. Contudo, antes de se proceder à detecção de obstáculos e reconhecimento de objectos é necessário efectuar um conjunto de procedimentos de maneira a definir cada uma das regiões de interesse e a quantificar a saliência de cada uma dessas regiões.

Para definir as regiões de interesse é necessário aplicar um filtro que crie um mapa de saliência, ou seja, uma imagem em que cada região está quantificada pelo seu grau de saliência

visual. Nesse mapa efectua-se depois a detecção e selecção das regiões individualizadas pela filtragem. A selecção consiste na escolha das regiões que tenham tamanho suficiente para poder conter informação útil, desprezando assim as regiões insignificantes. Depois de definidas as regiões, quantificam-se e ordenam-se as mesmas pela média dos valores de saliência dos seus pixéis. Este passo tem o objectivo de preparar as regiões para serem posteriormente processadas sequencialmente de acordo o seu grau de saliência (simulando assim o Foco-de-Atenção existente nos humanos). Só depois de devidamente definidas, quantificadas e ordenadas é que as regiões de interesse são processadas para detecção de obstáculos e reconhecimento de objectos utilizando inibição de retorno (IoR - *Inhibition of Return*). Cada um dos passos referidos será de seguida descrito de uma forma mais detalhada.

De maneira a efectuar a detecção das regiões mais salientes aplicou-se o algoritmo *Fast Saliency* [Butko et al., 2008] da biblioteca NMPT. Este algoritmo foi desenvolvido com a finalidade de se tornar numa aproximação a métodos mais populares, tais como o de [Itti et al., 1998], mas muito mais leve a nível de processamento, de modo a que pudesse ser utilizada em tempo real para direccionar as câmaras de um robô no desenvolvimento de robôs sociais (ver Secção 2.4). A principal característica deste algoritmo reside na sua rapidez de processamento, enquadrando-se assim nos requisitos necessários para a arquitectura descrita nesta dissertação.

O algoritmo é aplicado a todas as imagens (*frames*) captadas pelo robô, obtendo-se para cada uma delas uma outra imagem, $I_s(x, y)$, correspondente ao mapa de saliência. Um mapa de saliência consiste numa imagem em tons de cinzento em que as regiões mais salientes apresentam uma cor mais clara, e as menos salientes uma cor mais escura. A Figura 3.4 mostra um exemplo (imagem à direita) onde apenas as zonas que mais se destacam na imagem à esquerda (caixa cor de laranja, tampa da caixa preta, fita verde e os dois objectos que se encontram no chão) aparecem no mapa de saliência com tons mais claros, aparecendo tudo o resto com tons mais escuros.

De seguida é necessário que as regiões pequenas demais (insignificantes), tais como pixéis salientes individuais, sejam eliminadas. Para tal a imagem é separada em regiões quadradas (i), não sobrepostas, de tamanho $m \times m$ (com $m = 4$ para os resultados apresentados nesta dissertação). Em cada uma delas é contado o número de pixéis, N_s, cujo valor de saliência está acima de um valor mínimo Θ_1 (utilizou-se Θ_1 com o valor de 50% do máximo da saliência obtida na imagem I_s) e se o número de pixéis salientes for inferior a um valor limite Θ_2 (Θ_2

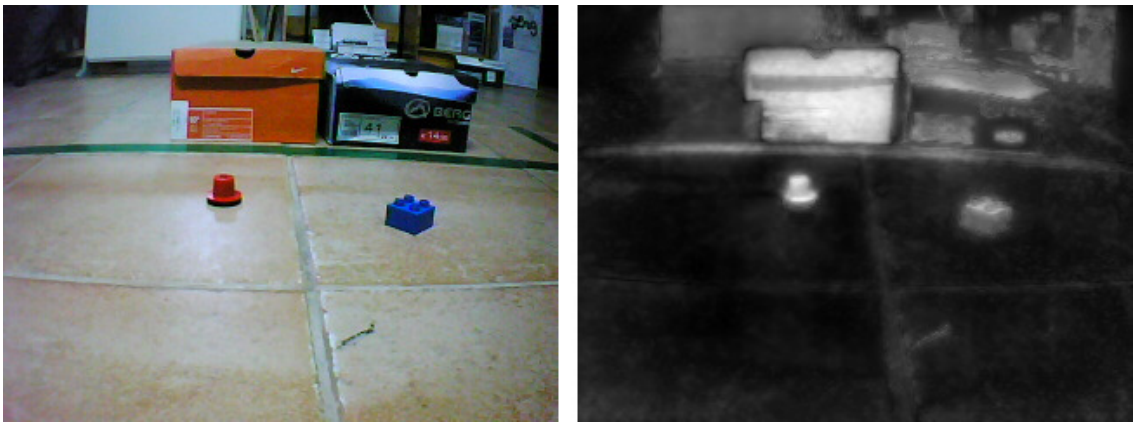

Figura 3.4: À esquerda a imagem capturada pela câmara do robô e à direita o mapa de saliência gerado a partir da mesma.

com o valor de 62,5% do total de pixéis existente em cada região), a região quadrada torna-se preta, caso contrário, mantém a sua tonalidade:

$$N_{s_i} = \sum_{(k,l)\in[0,m-1]} I_s(x+k, y+l), \text{ se } I_s(x+k, y+l) < \Theta_1, \tag{3.1}$$

$$i(x,y) = \begin{cases} 0 & \text{se } N_{s_i} \leq \Theta_2, \\ I_s(x,y) & \text{se } N_{s_i} > \Theta_2. \end{cases} \tag{3.2}$$

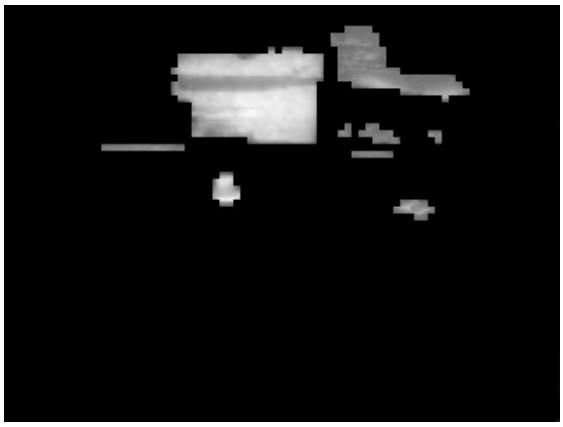

Figura 3.5: Mapa de saliência, I_{sf} após filtragem.

A perda de resolução na imagem resultante, I_{sf}, visível na Fig. 3.5 não provoca qualquer problema nos passos seguintes e, pelo contrário, até ajuda a que o processamento do passo seguinte, delimitação de regiões, seja bem mais rápido, pois em vez de ser necessário processar

$M \times N$ pixéis, basta processar um pixel por cada uma das regiões quadradas definidas, ou seja, $\frac{M}{4} \times \frac{N}{4}$ pixéis.

O processo de delimitação de regiões de interesse, ou detecção de *blobs* geralmente requer algum tempo de processamento, pelo que foi necessário a utilização de um método que reduzisse esse tempo de forma significativa. Utilizou-se assim uma variante do método descrito em [Saleiro et al., 2009], que consiste na delimitação das regiões de interesse por expansão das próprias regiões e é feito em três passos:

(a) Cada imagem I_{sf} é analisada linha a linha a partir do topo da imagem. Se três pixéis salientes (diferentes de 0) adjacentes forem encontrados numa linha, então é encontrado o início de uma região linear, ou *line-blob*. Após se ter encontrado o início de uma região linear, se três pixéis adjacentes não salientes (iguais a 0) forem detectados, então o fim dessa região é encontrado e as coordenadas mínimas e máximas em x são armazenadas em conjunto com a coordenada y da linha em que se encontram.

(b) Quando uma linha é processada, se existir uma região linear numa posição semelhante na linha anterior, então essa região é expandida para incluir a nova região linear.

(c) Após a detecção de todas as regiões, os seus tamanhos são calculados e eliminam-se as regiões cujo tamanho total seja inferior a um número mínimo de pixéis, Θ_3 (utilizou-se $\Theta_3 = 300$), resultando na imagem I_{sr}

Figura 3.6: Imagem com as regiões de interesse obtidas a partir do mapa de saliência, depois de filtradas (I_{sr}).

A Fig. 3.6 ilustra o resultado da filtragem sobre o mapa de saliência, onde se pode verificar que esta permite reduzir significativamente a área da imagem em que se efectuará o reconhecimento de objectos. Após a definição das regiões de saliência pode-se então proceder à detecção de obstáculos. Para tal, exploramos o mapa de saliência de modo a fazer uma detecção de onde o Foco-de-Atenção do robô primeiro vai estar.

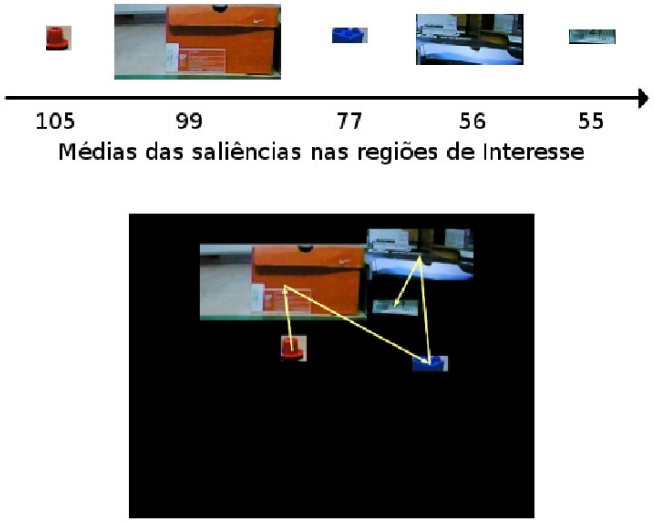

Figura 3.7: Em cima as regiões de interesse da imagem ordenadas por ordem decrescente das suas médias de saliência e em baixo o processo de inibição de retorno.

O Foco-de-Atenção é apenas aplicado a todas as imagens captadas quando a cabeça do robô se encontra direccionada na posição $P2$, ou seja quando a cabeça do robô se encontra mais levantada. Para determinar o ponto máximo de atenção calculam-se as médias da saliência das diversas regiões de interesse, ordenando-se as mesmas por ordem decrescente, de modo que o processo de reconhecimento de objectos seja feito primeiro nas regiões mais salientes, e só depois nas menos salientes (ver Fig. 3.7 no topo). Aplica-se também o processo de inibição de retorno, de forma a que uma região de saliência já explorada não volte novamente a ser explorada, (ver Fig. 3.7 em baixo, que ilustra através das setas a ordem pela qual o robô vai testar os objectos). Considerando cada uma das regiões ϕ, de tamanho $p \times q$, a média da saliência, N_e, nas mesmas é calculada utilizando os valores dos pixéis pertencentes às mesmas no mapa de saliência I_s:

$$N_{e_\phi} = \frac{\sum_{k\in[0,p-1],l\in[0,q-1]} I_s(x+k, y+l)}{p \times q}. \tag{3.3}$$

Após todo este processo de filtragem da informação útil da imagem e quantificação da importância dessa mesma informação, pode-se então proceder ao reconhecimento de objectos.

3.3 Reconhecimento de objectos

O reconhecimento de objectos é um procedimento que se realiza imediatamente a seguir à detecção do ponto máximo de atenção. Como referido anteriormente na Secção 3.1, o reconhecimento de objectos é feito utilizando a biblioteca OpenSURF. O método SURF é baseado no método SIFT, tendo como principais diferenças a sua maior rapidez e robustez [Evans, 2009] (ver Secção 2.5). Estes métodos consistem na detecção de pontos-chave invariantes a mudanças de escala e rotações nas imagens em que se pretende fazer o reconhecimento de objectos, sendo esses pontos-chave comparados com vectores de pontos-chave extraídos de imagens previamente conhecidas.

Da mesma forma que nós, humanos, guardamos na nossa memória imagens normalizadas de um grande número de objectos que servem depois para efectuar o reconhecimento dos mesmos, o robô também necessita de já ter na sua memória uma ou mais imagens dos objectos que poderá encontrar no seu percurso. Todas as imagens que são previamente conhecidas pelo robô são capturadas a cerca de $45cm$ de distância (exemplos na Fig. 3.8), havendo assim a possibilidade de o robô saber um valor aproximado (salienta-se aproximado e não exacto, pois o mesmo o método SURF permite alguma relaxação em termos de dimensões dos objectos a comparar) da distância a que os mesmos se encontram a partir das dimensões dos objectos reconhecidos.

Figura 3.8: Exemplo de alguns objectos previamente dados ao robô.

No início da execução do programa do robô, todos os objectos que são previamente forne-

cidos passam pelo processo de reconhecimento de pontos-chave (ver Secção 2.5), ficando os seus pontos-chave armazenados em vectores, de modo que, durante a execução do programa, a sua comparação com os vectores obtidos a partir das imagens captadas pelo robô seja feita de forma mais rápida. Mais uma vez, o que guardamos são alguns pontos característicos, não a imagem em si, uma vez que o que o nosso cérebro faz é guardar alguns atributos que representam o objecto [Rodrigues and du Buf, 2009a,b]. No futuro o objectivo será trocar o método de reconhecimento de objectos SURF por um método completamente biológico de reconhecimento de objectos tal como o proposto por Rodrigues e du Buf [2009b] baseado em linhas, arestas e pontos-chave.

A biblioteca OpenSURF permite que a sensibilidade para detecção de pontos-chave seja calibrada, sendo que o processamento tende a ficar mais pesado com o aumento do número de pontos-chave. Na Fig. 3.9 pode verificar-se a diferença entre dois valores de sensibilidade aplicados sobre a mesma imagem.

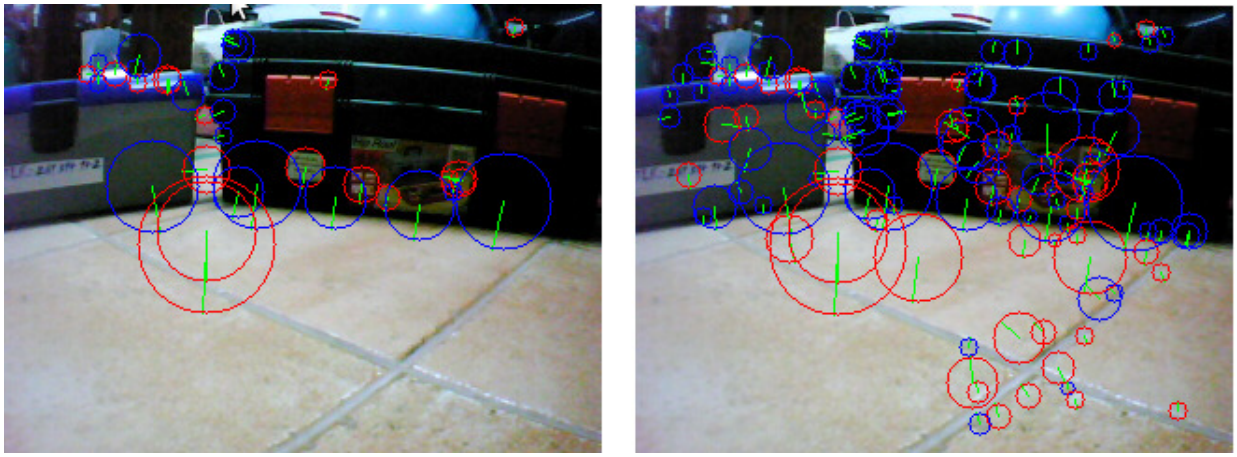

Figura 3.9: Resultado de diferentes calibrações na sensibilidade de detecção de pontos-chave. Na imagem à esquerda tem menos sensibilidade e à direita tem mais sensibilidade [Lowe, 2004].

Quando se procede à comparação dos vectores de pontos-chave dos objectos previamente conhecidos com os vectores das imagens adquiridas, não é necessário que exista uma correspondência para cada um dos pontos-chave, podendo-se considerar que o reconhecimento de um objecto é feito com sucesso a partir do momento em que se encontrem 3 correspondências, tal como acontece no método SIFT. Na Figura 3.10 pode-se visualizar a existência de 5 correspondências entre a imagem previamente fornecida e a imagem captada pelo robô. Pode-se

verificar que apesar das distâncias, das perspectivas de visão e das condições de iluminação diferentes é possível efectuar o reconhecimento de objectos com alguma robustez e com rapidez suficiente para aplicações em tempo real. Mais uma vez, este processo tem semelhança com o que o cérebro faz, sendo ainda alvo de investigação quantos pontos (atributos) e vistas devem ser guardadas para cada objecto [Rodrigues and du Buf, 2009b].

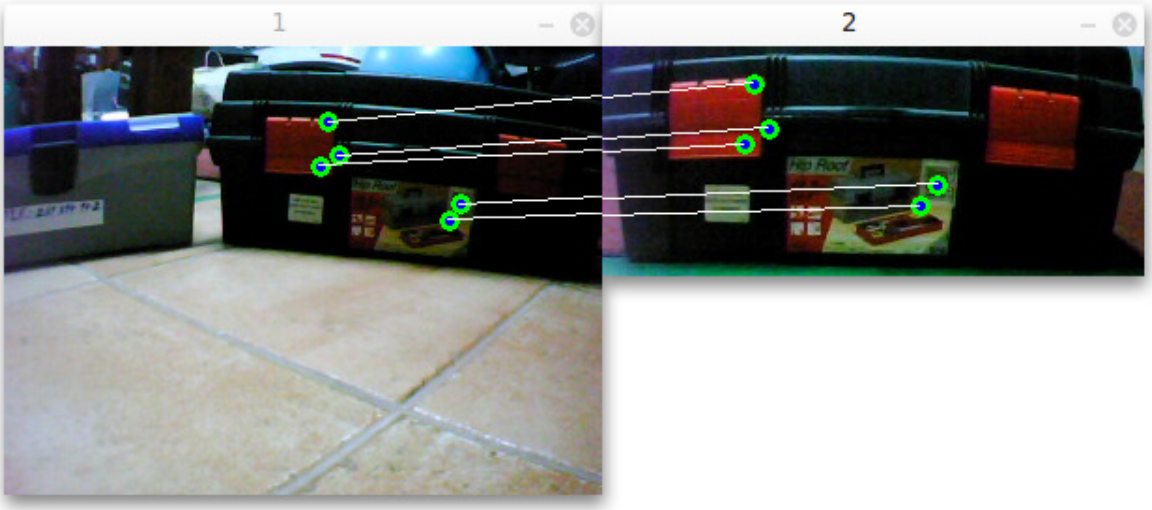

Figura 3.10: Exemplo do reconhecimento de um objecto, evidenciando as 5 correspondências de pontos-chave entre as duas imagens.

O reconhecimento de objectos e a saliência compõem o sistema visual do robô que recolhe toda a informação necessária à navegação do robô e à realização de tarefas. Para que o robô consiga realizar tarefas é necessária a existência de uma estrutura que permita combinar a recolha de informação do sistema visual com acções motoras. Na Secção seguinte será feita a descrição dessa estrutura.

3.4 Gestão de tarefas

Tal como referido na Secção 2.6, outro dos grandes desafios da robótica cognitiva é a implementação de uma estrutura de tarefas que permita a aprendizagem e a construção de tarefas complexas a partir de tarefas mais simples. A estrutura implementada neste trabalho é semelhante à das propostas de [Ratanaswasd et al., 2005; Alami et al., 2006] no sentido em que existe um Agente Central que é responsável por construir e organizar as tarefas de modo a cumprir o objectivo solicitado.

Considera-se que existem dois tipos principais de tarefas: as micro e as macro-tarefas. As micro-tarefas correspondem a acções básicas tais como "virar para a direita," "andar para a frente," "olhar para a frente" ou "adquirir informação visual." As macro-tarefas são depois construídas com base na agregação de micro-tarefas em ciclos de acção. Por sua vez, as macro-tarefas são já tarefas mais complexas, mas que podem ter vários níveis de complexidade, pois para além das macro-tarefas construídas a partir de micro-tarefas, podemos também ter macro-tarefas construídas a partir de outras macro-tarefas.

As macro-tarefas mais simples têm que ser "ensinadas ao robô" pelo operador e estão sempre associadas a uma acção simples tal como "procurar," "contar," "regressar," etc. Posteriormente quando é dada uma tarefa ao robô que envolva essas macro-tarefas já conhecidas, tal como "procura objecto e regressa," essas macro-tarefas são colocadas num *buffer* de planeamento de tarefas para que sejam executadas. Para que este sistema seja funcional, é necessário que cada macro-tarefa seja composta por três blocos.

O primeiro consiste no (a) ciclo de acções visio-motoras, ou seja, consiste no ciclo de acções que envolve as acções motoras que são feitas em função da informação visual captada. O segundo (b) consiste na função de detecção do objectivo pretendido. Por fim, o terceiro (c) consiste na condição de verificação de tarefa completa ou incompleta.

Os blocos são separados para que possam ser realizadas várias tarefas ao mesmo tempo. Em qualquer momento existe sempre uma tarefa principal, sendo o seu bloco visio-motor o único em funcionamento, deixando todos os blocos visio-motores de outras tarefas em espera. No entanto, enquanto esse bloco visio-motor mantém o controlo do robô, nada impede que as funções de detecção para a realização dos objectivos das outras tarefas sejam realizadas. De uma forma simples, se for ordenado ao robô que vá até ao local onde se encontra o objecto X e conte até cinco objectos do tipo Y, nada impede que o robô procure ao mesmo tempo por objectos dos dois tipos e procure objectos do tipo Y durante o seu trajecto até ao local onde se encontra o objecto X, apesar de apenas o bloco visio-motor de uma das tarefas poder manter o controlo do robô de cada vez.

Quando uma tarefa é concluída, o seu bloco visio-motor é desactivado e o bloco da tarefa seguinte é activado, e assim sucessivamente até que todas as tarefas sejam realizadas.

Existem 3 *buffers* de blocos separados: um de blocos visio-motores; outro de blocos de detecção; e outro de blocos de verificação de conclusão de tarefas. Os blocos destes dois últimos *buffers* são executados sequencialmente dentro de cada ciclo do bloco visio-motor que

mantenha o controlo do robô. Dos três tipos de blocos, os blocos visio-motores são sempre os mais complexos, pois os outros dois tipos de blocos traduzem-se em funções simples de detecção ou verificação de condições. Os que são responsáveis pela detecção necessitam de ter sempre como entrada o objecto a reconhecer, tendo este que existir na biblioteca de objectos conhecidos.

Sempre que uma tarefa é concluída, os blocos correspondentes são marcados como inactivos e o robô passa a executar o novo bloco visio-motor e os blocos de detecção e verificação de conclusão das restantes tarefas.

Cada tarefa, para além dos três blocos, tem ainda associada uma palavra chave que representa a acção a executar. Por exemplo ao dar o comando "procura" e "regressa" o robô irá colocar os blocos correspondentes às duas tarefas nos *buffers* correspondentes.

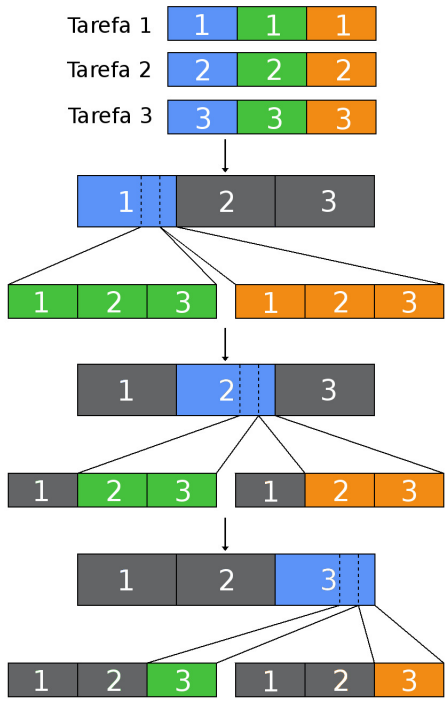

Figura 3.11: Exemplo do processo de construção e execução de uma macro-tarefa a partir de três macro-tarefas mais simples. Os blocos azuis correspondem aos blocos visio-motores, os verdes aos blocos de detecção e os laranjas aos blocos de verificação de conclusão de tarefas, ver texto para mais detalhes.

Independentemente das tarefas que estejam a ser realizadas, as imagens captadas são todas processadas para reconhecimento de objectos de referência, de modo a que o robô

possa corrigir a sua posição. Em simultâneo é também guardado um *buffer* com os objectos de referência por onde o robô vai passando, de forma a que possa posteriormente saber o caminho até qualquer um deles.

Na Fig. 3.11 está representado o processo de construção e execução de uma macro-tarefa a partir de três outras micro- e macro-tarefas mais simples. A azul estão representados os blocos visio-motores, a verde os blocos de detecção e a laranja os blocos de verificação de conclusão de tarefas. Os blocos a cinzento são blocos que estão inactivos. Cada uma das tarefas 1, 2 e 3 são compostas por um bloco de cada tipo. A construção da tarefa mais complexa é feita através da separação dos blocos das tarefas mais simples, colocando os blocos de cada tipo num *buffer* diferente. Os blocos visio-motores são executados um de cada vez, só passando para o seguinte quando o anterior é concluído. Por sua vez, os blocos contidos nos *buffers* dos outros dois tipos são executados sequencialmente em cada ciclo do bloco visio-motor em acção. Esta estrutura permite ir fazendo a detecção de objectos que sejam necessárias a outras tarefas apesar de as suas acções visio-motoras serem direccionadas para realizar uma outra tarefa principal.

Na primeira linha da imagem encontram-se as três tarefas em separado, que são depois decompostas nos sus três blocos, sendo esses blocos colocados nos respectivos *buffers* de execução. Na segunda linha temos os três *buffers* com os blocos correspondentes, estando os blocos visio-motores 2 e 3 inactivos pois o bloco visio-motor 1 mantém o controlo do robô. Na terceira linha todos os blocos relativos à tarefa 1 são desactivados, pois a tarefa foi concluída e o bloco visio-motor 2 assume o controlo do robô. Por fim, quando a tarefa 2 é concluída, todos os seus blocos são desactivados e o bloco visio-motor 3 passa a comandar o robô.

Para a navegação do robô considerou-se que existem dois modos fundamentais: (a) o **modo de exploração** que permita navegar num ambiente desconhecido; e (b) o **modo de excursão** que permita fazer uso da informação recolhida para se deslocar de forma mais eficiente.

Ambos os modos de navegação são implementados sob a forma de blocos visio-motores que podem ser depois utilizados para a realização de diversas tarefas, bastando para isso associá-los a diferentes blocos de detecção e verificação de conclusão de tarefa. Em seguida, far-se-á uma descrição de ambos os modos de navegação.

3.4.1 Modo de exploração

No modo de exploração a saliência visual tem um papel muito importante, pois é com base na saliência que o robô decide para onde deve ir, pois nas áreas de maior saliência é mais provável que exista informação importante para o robô. Contudo, antes de se fazer qualquer uso dos mapas de saliência das imagens captadas para tomar qualquer decisão, estes são processados para que a saliência proveniente da fita verde (limitadora do ambiente) seja reduzida.

Para tal procede-se inicialmente à detecção da fita (limites do ambiente), obtendo-se uma imagem binária em que a fita verde aparece a branco e tudo o resto a preto. Após este primeiro passo, a imagem é processada pixel a pixel e sempre que um pixel branco for encontrado, o pixel correspondente no mapa de saliência será colocado a preto, reduzindo assim o efeito da saliência provocado pela fita. O algoritmo de detecção da fita é descrito na Secção 3.5 que descreve o sistema de localização e mapeamento simultâneos.

A primeira componente visio-motora deste modo reside nos movimentos da cabeça do robô. Inicialmente o robô olha em frente na perspectiva P2 e divide horizontalmente a imagem captada em três regiões de igual tamanho, uma à esquerda, outra ao centro e outra à direita. Em seguida calculam-se as médias da saliência nas regiões da esquerda e da direita e se as médias forem superiores a um valor limite T_s (utilizou-se T_s com 14% do valor máximo da saliência em T_s) o robô olha para a próxima posição à esquerda ou para a próxima posição à direita, olhando primeiro para o lado que tiver maior média de saliência. No caso de a cabeça do robô se encontrar direccionada para a primeira posição à esquerda, a imagem captada nesta posição é também dividida em três zonas mas só a média da saliência da zona à esquerda será calculada, servindo para decidir se vale a pena rodar a cabeça para a segunda e última posição à esquerda. O mesmo acontece quando a cabeça do robô se encontra direccionada para a primeira posição à direita, sendo calculada apenas a saliência da região da direita. Esta componente permite assim direccionar a cabeça do robô para as regiões com maior interesse.

O passo seguinte consiste em calcular os valores heurísticos que permitem ao robô qual o rumo a tomar: se deve virar à esquerda; se deve ir em frente; ou se deve virar à direita. Para tal, divide-se o campo de visão composto pelas cinco imagens em três regiões, tal como representado pela Fig. 3.12.

A azul está representada a região esquerda, a verde a frontal e a laranja a região direita.

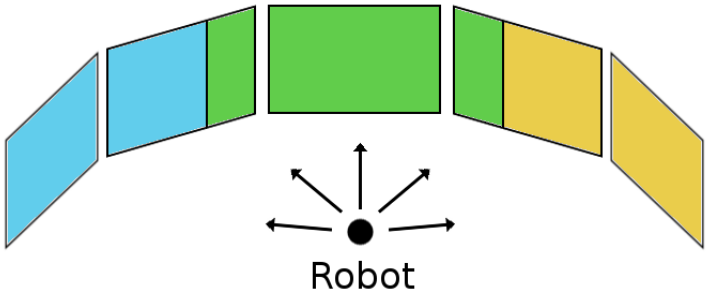

Figura 3.12: Representação das três zonas compostas pelas imagens captadas nas cinco perspectivas em P2.

Há que realçar que no caso de o robô não ter olhado para alguma das direcções, não tendo captado a imagem correspondente, a saliência é considerada como sendo zero. De seguida calculam-se as médias das saliências das três regiões, esquerda (s_e), frontal (s_f) e direita (s_d) e posteriormente a saliência relativa (s_{x_r}) de cada uma delas, com $x = \{e, f, d\}$:

$$s_{x_r} = \frac{s_x}{s_e + s_f + s_d} \text{ com } x \in \{e, f, d\}. \tag{3.4}$$

Um outro factor importante para determinar que direcção deve seguir é a distância das novas possíveis posições às posições ocupadas anteriormente ou aos locais de referência já visitados, de forma a que não se volte a locais já explorados se ainda houver outros locais por explorar.

Se já tiver sido encontrado um local de referência anteriormente, calcula-se a distância entre esse mesmo local e as três possíveis posições que poderão ser assumidas pelo robô no final da próxima acção motora, d_e, d_f, d_d (se se deslocar para a esquerda passará a estar $25cm$ à esquerda da posição actual, se se deslocar para a frente estará $25cm$ à frente da posição actual e se se deslocar para a direita estará $25cm$ à direita). Caso ainda não tenha sido encontrado nenhum local de referência, calcula-se a distância à terceira última posição ocupada pelo robô. Por fim, tal como se fez para a saliência, calcula-se a distância relativa de cada uma delas, d_{x_r}, com:

$$d_{x_r} = \frac{d_x}{d_e + d_f + d_d} \text{ com } x \in \{e, f, d\}. \tag{3.5}$$

Para finalmente determinar o potencial (P) de cada uma das três opções (esquerda,

direita ou frente) utilizam-se as expressões seguintes:

$$P_{e/d} = (1 - k) \times d_{x_r} + k \times s_{x_r} \text{ com } w \in [0, 1], \tag{3.6}$$

$$P_f = w \times ((1 - k) \times d_{x_r} + k \times s_{x_r}) \text{ com } k \in [0, 1] \text{ e } k \in [0, 2] \tag{3.7}$$

O parâmetro k serve para que se possa escolher a importância a dar à componente da saliência em relação à componente da distância. Se for um valor próximo de zero o movimento do robô é feito de forma a deslocar-se para mais longe do último objecto conhecido resultando numa exploração mais rápida. Por outro lado, se for próximo de 1, o movimento é feito dando mais importância à saliência visual, resultando numa exploração mais lenta, mas mais detalhada. O potencial da opção de andar para a frente, P_f, é calculado de maneira diferente dos potenciais para virar à esquerda, P_e, ou direita, P_d, pois é afectado do parâmetro w, que serve para poder regular a tendência do robô em seguir em frente ou virar frequentemente. Nos testes utilizaram-se os valores $k = 0.25$ e $w = 1.65$. Depois de determinada a direcção com maior potencial adquirem-se as imagens na posição P1 na direcção para onde se pretende virar e coloca-se a informação relativa a obstáculos ou limites na memória a curto prazo.

Na memória a curto prazo faz-se depois a consulta pela existência de obstáculos na direcção pretendida. Caso não exista nada que impeça, o robô segue o caminho escolhido. Caso contrário, faz-se a mesma análise para o segundo caminho com mais potencial, e assim sucessivamente.

O processo descrito nesta Secção permite fazer a exploração do meio ambiente e vai-se repetindo ciclicamente até que o objectivo seja cumprido.

3.4.2 Modo de excursão

Este modo de navegação é relativamente mais simples do que o de exploração, pois neste caso o robô já sabe as localizações de alguns pontos de referência e tem na sua memória as experiências passadas, podendo utilizá-las para saber o percurso e tudo o que tem que fazer é percorrer o caminho inverso ao que foi feito durante o período de exploração. Quando aparecem obstáculos no caminho do robô, captam-se as imagens à esquerda e à direita utilizando a perspectiva P1 afim de determinar qual dos caminhos está livre. Se ambos estiverem livres o robô vira para o que resultar numa menor distância ao objectivo seguinte.

Assim que for possível voltar a seguir na mesma direcção que seguia anteriormente, o robô volta a virar para o lado contrário e segue o seu caminho. Mais uma vez, este processo assemelha-se ao que o ser humano faz no seu dia a dia.

3.5 Localização e mapeamento simultâneo

O sistema de navegação e mapeamento simultâneo é composto pela agregação de diversas componentes referidas nas Secções anteriores (visão, memória e acções motoras), havendo assim um ciclo que se repete continuamente tal como acontece connosco, humanos: a informação adquirida pela visão é processada, passando depois para a memória, de forma a poder ser usada para tomar a decisão de qual a acção motora a realizar. Deste modo, o sistema de SLAM descrito inspira-se assim no sistema humano e como tal, não necessita de sistemas de odometria de precisão.

Como já foi referido anteriormente, apenas se utiliza uma das câmaras do robô, que em conjunto com o sistema de rotação e inclinação da cabeça adquire a informação do meio envolvente. Apenas se utilizam dois ângulos de inclinação da cabeça: um deles permite observar o ambiente mais próximo do robô com algum detalhe e o outro permite observar o ambiente um pouco mais afastado, dando assim um maior campo de visão e menos detalhe. Os dois ângulos serão designados como $P1$ e $P2$, respectivamente (ver Secção 3.1).

Todas as imagens captadas com $P1$ ou $P2$ são processadas para detecção de obstáculos (objectos no meio do percurso ou fita limitadora do ambiente). Após o processamento para detecção de obstáculos, são calculadas as distâncias aos mesmos com uma função interpoladora que relaciona cada linha de uma imagem a uma distância pouco precisa. Depois de se calcularem as distâncias aos objectos vão sendo criados dois mapas: (a) um mapa mais pequeno mas com maior precisão (corresponde à memória a curto prazo); e (b) um mapa maior e com menos precisão (corresponde à memória a longo prazo).

O primeiro, (a) corresponde à memória que é usada para decisões imediatas e o segundo, (b) corresponde à memória que é usada para a realização de tarefas mais complexas. Os dois mapas têm ainda algumas outras diferenças entre si, tais como a forma como a informação é armazenada, o tipo de informação registada e a duração da informação neles contida. Os dois tipos de memória e as características dos mapas serão descritas em mais detalhe mais à frente. Para além das distâncias aos obstáculos encontrados, nas imagens obtidas com $P2$

procede-se ainda ao reconhecimento de objectos com o método já descrito na Secção anterior, que servem depois para que o robô possa recalibrar a sua posição e orientação.

3.5.1 Detecção de obstáculos e limites

A detecção de obstáculos e limites consiste num processo fundamental na navegação do robô, pois só assim se pode determinar em que partes do ambiente envolvente é que o robô pode circular. Todas as imagens adquiridas pelo robô passam pelo processo de detecção de obstáculos e limites, sendo os primeiros detectados através da saliência e os segundos detectados pela sua cor verde, pois o ambiente do robô é delimitado com fita verde no chão.

A detecção da fita verde é um processo bastante trivial, começando com a conversão das imagens $I(x, y)$ do espaço de cor RGB (*Red, Green, Blue*) para o espaço de cor HSV (*Hue, Saturation, Value*), dando origem a $I_{HSV}(x, y)$. Esta conversão tem por objectivo tornar mais fácil limitar o intervalo de cores em que a fita verde se encontra. A conversão para HSV é feita pixel a pixel, sendo cada pixel P_{RGB}, composto pelas componentes R, G e B, convertido para um pixel P_{HSV}, com as componentes H, S, V, utilizando as seguintes expressões [Gonzalez and Woods, 2007]:

$$M = max(R, G, B); \tag{3.8}$$

$$m = min(R, G, B); \tag{3.9}$$

$$\Delta = M - m; \tag{3.10}$$

$$H = \begin{cases} indefinido & \text{se } \Delta = 0, \\ 60^{\text{o}} \times \frac{G-B}{\Delta} & \text{se } M = R, \\ 60^{\text{o}} \times (\frac{B-R}{\Delta} + 2) & \text{se } M = G, \\ 60^{\text{o}} \times (\frac{R-G}{\Delta} + 4) & \text{se } M = B; \end{cases} \tag{3.11}$$

$$V = M; \tag{3.12}$$

$$S = \begin{cases} 0 & \text{se } \Delta = 0, \\ \frac{\Delta}{V} & \text{se } \Delta \neq 0. \end{cases} \tag{3.13}$$

Após a conversão das imagens para HSV, estas são analisadas de forma a manter na imagem apenas as zonas correspondentes à fita verde. Através da análise das cores correspondentes à fita verde em diversas imagens, verificou-se que o valor de H se situava entre 100 e 190, e o valor de S entre 0.6 e 1. Todos os pixéis com valores compreendidos entre os valores referidos passam a ter o valor 255 (branco) e os restantes passam a ter o valor 0 (preto), ficando-se assim com uma imagem binária, $I_f(x, y)$ (ver Fig. 3.13 na segunda linha, à esquerda), em que os pixéis com a cor da fita estão a branco, e todos os restantes estão a preto.

$$I_f(x,y) = \begin{cases} 255 & \text{se } 100 < H_{I_{HSV}(x,y)} < 190 \text{ e } 0,6 < S_{I_{HSV}(x,y)} < 1, \\ 0 & \text{outros casos.} \end{cases} \tag{3.14}$$

Por fim faz-se uma filtragem na imagem, de forma a homogeneizar as zonas correspondentes à fita verde, eliminando ao mesmo tempo eventuais pixéis ou pequenas regiões isoladas que tenham sido consideradas como sendo fita verde devido a variações na iluminação ou erros na câmara. A filtragem aplicada é semelhante à que é aplicada no processo saliência. A imagem é sucessivamente separada em regiões quadradas Ψ e sobrepostas de tamanho $n \times n$ (utilizou-se $n = 6$). Em cada uma delas é contado o número de pixéis brancos, N_f, e se o número de pixéis brancos for inferior a um valor limite Θ (com Θ igual a 28% do total de pixéis existentes na região), a região quadrada torna-se preta. Caso contrário, torna-se branca.

$$N_{f_\Psi} = \sum_{(k,l)\in[0,m-1]} I_f(x+k, y+l), \text{ se } I_f(x+k, y+l) = 255, \tag{3.15}$$

$$\Psi_{I_{ff}}(x,y) = \begin{cases} 0 & \text{, se } N_{f_\Psi} < \Theta, \\ 255 & \text{, se } N_{f_\Psi} \geq \Theta. \end{cases} \tag{3.16}$$

Na segunda linha da Figura 3.13 à esquerda pode ver-se a imagem resultante, I_{ff}, do processo de detecção de limites do ambiente (fita) que delimita o ambiente em que o robô pode navegar.

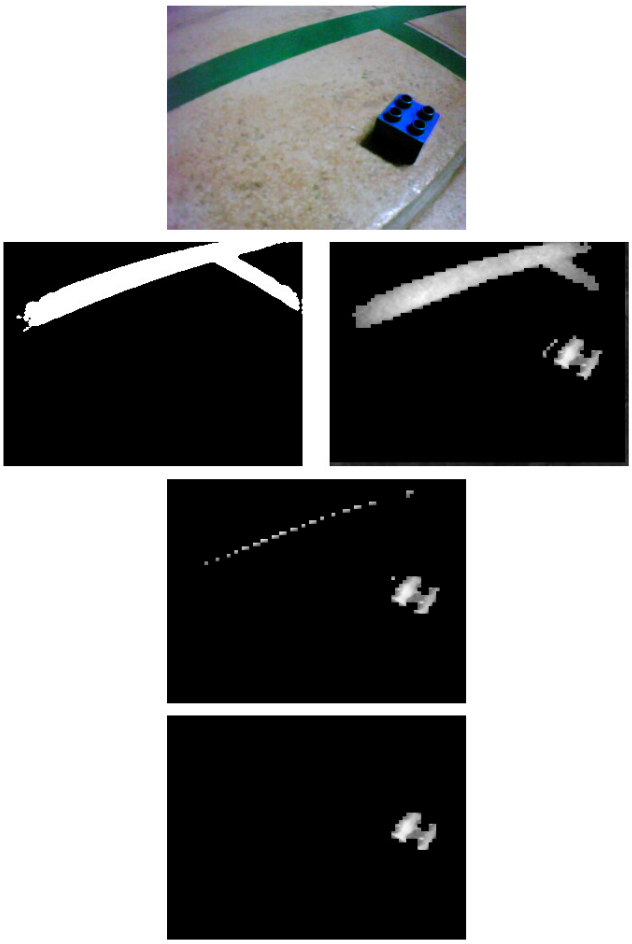

Figura 3.13: Processo de detecção de obstáculos: em cima a imagem captada I, com fita verde e um obstáculo; na segunda linha à esquerda a imagem resultante da detecção da fita verde, I_{ff}, e à direita a imagem resultante do processo de detecção de saliência, I_{sf}; na terceira linha a imagem resultante da remoção de zonas de fita verde, I_o; por fim a imagem apenas com o obstáculo, I_{of}.

No que diz respeito à detecção de obstáculos utiliza-se a imagem resultante do processo de detecção de zonas salientes antes de se proceder à delimitação dessas mesmas regiões (ver Figura 3.5). Todos os pixéis dessa imagem, I_{sf} (ver Fig. 3.13 na segunda linha, à direita), que não sejam pretos e que coincidem com a cor branca da imagem da detecção da fita verde (I_{ff}) são considerados como zona de fita e são eliminados, resultando na imagem I_o (ver Fig. 3.13, terceira linha). Para remover as regiões insignificantes em I_o, divide-se tanto I_{ff} como I_{sf} em regiões quadradas (γ) de tamanho $m \times m$ e conta-se o número de pixéis brancos (N_o) em comum entre cada região de I_{ff} e a região correspondente em I_{sf}. Obtemos assim

a imagem final com o obstáculo e sem fita I_{of} (ver Fig. 3.13, quarta linha):

$$N_{o_\gamma} = \sum_{(k,l)\in[0,m-1]} I_o(x+k, y+l), \text{ se } \gamma_{I_{sf}}(x+k, y+l) = 255 \text{ e } \gamma_{I_{ff}}(x+k, y+l) = 255, \tag{3.17}$$

$$\gamma_{I_{of}}(x,y) = \begin{cases} 0 & \text{se } N_o \geq 1, \\ 255 & \text{se } N_o < 1. \end{cases} \tag{3.18}$$

De seguida é necessário converter a informação proveniente das imagens binárias da I_{ff} e I_{of} em distâncias e ângulos aproximados para a posterior criação de mapas e armazenamento em memória. Uma vez que não se necessita de grande precisão nas medidas criaram-se funções de aproximação à distância em função da linha da imagem. Foram criadas duas funções, uma para a posição P1 e outra para a posição P2. Para a criação de cada função utilizou-se um conjunto de 4 pontos adquiridos por intermédio de testes práticos e apresentados na Tabela 3.1, sobre os quais se aplicou o polinómio de Newton para criar funções interpoladoras [Zarowski, 2004]:

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + ... + f[x_0, ..., x_n](x - x_0)...(x - x_{n-1}) \tag{3.19}$$

em que:

$$f[x_i, ..., x_{i+k}] = \frac{f[x_{i+1}, ..., x_{i+k}] - f[x_i, ..., x_{i+k-1}]}{x_{i+k} - x_i}, \tag{3.20}$$

tendo resultado as equações 3.21 e 3.22, para P1 e P2, respectivamente:

$$D_{P1} \simeq 4.5 \times 10^{-6} \times x^3 + 2.133 \times 10^{-3} \times x^2 + 0.387 \times x + 51.143; \tag{3.21}$$

$$D_{P2} \simeq 7.129 \times 10^{-5} \times x^3 + 0.037 \times x^2 + 6.314 \times x + 420.957. \tag{3.22}$$

Com estas equações torna-se então possível obter as distâncias aproximadas aos limites do ambiente de navegação do robô ou aos obstáculos que se encontrem no seu caminho. Contudo a distância não é suficiente para localizar um ponto na imagem, pelo que é necessário calcular

P1		P2	
Linha	Distância[cm]	Linha	Distância[cm]
3	50	72	130
35	40	96	90
92	30	144	60
225	20	234	40

Tabela 3.1: Correspondência entre linhas das imagens e as respectivas distâncias para as posições P1 e P2

também o ângulo em relação ao robô. Para tal, calculou-se a largura do campo de visão, W, à distância, d, a que o ponto se encontra:

$$W = 2 \times d \times tg(\frac{FOV}{2}) \tag{3.23}$$

em que $FOV(Field\ of\ View)$ é o ângulo de visão da câmara, que é de 65º. De seguida é necessário calcular a distância do ponto a um ponto que se encontre na mesma linha, mas mesmo em frente ao robô, ou seja, um ponto cujo ângulo em relação ao robô seja de 0º. Para tal foi necessário definir uma recta que representa todos os pontos que se encontram mesmo em frente ao robô: colocou-se o robô centrado com uma linha recta no chão e determinaram-se depois dois pontos dessa linha recta, de modo a poder definir essa mesma recta por uma equação. O processo teve que ser feito tanto para P1 como para P2.

Na Tabela 3.2 encontram-se os quatro pontos utilizados para definir as rectas (dois pontos para cada recta).

P1		P2	
Linha	Coluna	Linha	Coluna
8	188	1	135
320	284	318	235

Tabela 3.2: Pontos utilizados para definir as rectas que fazem um ângulo de 0º em relação ao robô.

Utilizando os pontos referidos definiram-se equações para as duas rectas, sendo c a coluna

e l a linha:

$$c_{P1} = \frac{l_{P1} + 603}{3,25}, \tag{3.24}$$

$$c_{P2} = \frac{l_{P2} + 427}{3,17}. \tag{3.25}$$

Utilizando estas equações pode-se então determinar numa linha da imagem a coluna, c, que corresponde a uma posição frontal ao robô. De seguida, para obter a distância D_l entre o ponto em que o obstáculo se encontra $P(x, y)$ e a recta, calcula-se o número de pixéis existentes entre ambos, na mesma linha da imagem, multiplicando-se depois pela distância por pixel, D_p, que é obtida ao dividir a largura do campo de visão, W, pela largura da imagem em pixéis, L,

$$D_p = W/L; \tag{3.26}$$

$$D_l = |c - x_P| * D_p. \tag{3.27}$$

Por fim, calcula-se o ângulo α do ponto P(x, y) em relação ao robô utilizando o teorema de Pitágoras:

$$\alpha = \begin{cases} tg^{-1}(\frac{D_l}{d}) & \text{se } x_P > c, \\ -tg^{-1}(\frac{D_l}{d}) & \text{se } x_P < c. \end{cases} \tag{3.28}$$

A este ângulo soma-se ainda o ângulo de rotação da cabeça do robô quando a imagem é tirada. Como referido no Capítulo anterior, são utilizadas apenas cinco posições de rotação. A Tabela 3.3 apresenta os ângulos correspondentes às cinco posições de rotação da cabeça.

Tendo o ângulo e a distância a qualquer ponto na imagem pode-se então localizar espacialmente num sistema de coordenadas polares qualquer informação relevante que esteja presente nas imagens adquiridas, podendo-se assim criar mapas do ambiente envolvente.

O método de cálculo da distância e do ângulo não tem grande precisão e não tem em conta factores como a distorção da imagem provocada pela lente da câmara, mas proporciona a possibilidade de se conseguirem criar referências espaciais de forma rápida, simples e sem grandes exigências a nível de processamento. Tal como foi referido anteriormente o objectivo

Posição Horizontal	Ângulo
-2H	-52º
-1H	-26º
0H	0º
1H	26º
2H	52º

Tabela 3.3: Ângulos correspondentes às cinco posições de rotação da cabeça.

é ter apenas referências espaciais pouco precisas, pois tendo como exemplo o sistema visual humano, a precisão nas distâncias não é essencial à navegação.

Concluindo, nesta Secção descreveram-se os algoritmos de processamento de imagem que são utilizados para converter a informação visual em informação útil para o robô. Na Secção seguinte será feita uma descrição das estruturas de memória necessárias para armazenar e utilizar essa informação.

3.5.2 Memória e mapeamento

A construção de mapas do ambiente é fundamental para que o robô possa navegar e localizar-se espacialmente. Para construir um mapa é necessário escolher a informação importante para o mesmo e armazená-la em memória. Contudo, como referido no segundo Capítulo desta dissertação, se considerarmos a memória humana como exemplo verificamos que a própria organização da memória engloba uma filtragem da informação, sendo que partes da informação recolhida nem chegam a entrar na memória a curto prazo, outras não passam da memória a curto prazo para a memória a longo prazo, outras permanecem pouco tempo na memória a longo prazo e outras partes permanecem na memória a longo prazo por longos períodos de tempo. Seguindo o modelo da memória humana definiu-se que a memória do robô seria também bipartida em memória a curto prazo e memória a longo prazo, que designaremos de STM e LTM, respectivamente (ver Secções 2.2 e 2.3).

Cada um dos dois tipos de memória corresponde a um mapa de características diferentes. A STM consiste num mapa de reduzidas dimensões que tem como único objectivo registar toda a informação recolhida nas proximidades do robô que possa servir para tomar decisões no que diz respeito à navegação imediata. Um exemplo dessa informação é a presença de

fita delimitadora do ambiente ou de obstáculos.

Como tal, para a construção deste mapa apenas se utilizam as imagens captadas com a cabeça do robô na posição $P1$, em qualquer uma das cinco posições horizontais da cabeça. A STM é relativa à posição corrente do robô agregando num mapa toda a informação recolhida enquanto o robô permanecer nessa mesma posição. Este mapa é bastante pequeno (I_{STM}) e tem uma resolução maior que a LTM. Para a construção de um mapa as imagens binárias de tamanho $M \times N$ obtidas através do processo descrito na Secção anterior são analisadas de 20 em 20 colunas e pixéis, de baixo para cima.

Sempre que um pixel branco é encontrado nas imagens I_ff e I_of utilizam-se as equações referidas na Secção anterior para obter a posição do mesmo em relação ao robô em coordenadas polares. O mapa é guardado sob o formato de uma imagem de tamanho 100×50 pixéis, de forma a que 1 pixel corresponde a $1cm$. O facto de o mapa ter 50 linhas deve-se à distância de visão máxima para a posição P1 ser de $50cm$. As coordenadas polares são então convertidas para coordenadas cartesianas (l, c), sendo depois o pixel correspondente a essas coordenadas colocado a preto. Para converter as coordenadas polares, $D.cis(\Theta)$, para cartesianas utilizaram-se as seguintes equações:

$$c = \frac{N}{2} + sin(\Theta) \times D, \tag{3.29}$$

$$l = M - cos(\Theta) \times D. \tag{3.30}$$

De seguida divide-se a imagem em regiões quadradas Ψ e não sobrepostas de tamanho $m \times m$. Se houver um pixel preto numa região, então a mesma torna-se preta. Caso contrário, mantém-se branca. A I_{STM} obtém-se da seguinte forma:

$$N_{STM} = \sum_{(k,l)\in[0,m-1]} I_{fo}(x+k, y+l), \text{ se } I_{fo}(x+k, y+l) = 0, \tag{3.31}$$

$$\Psi_{I_{STM}}(x,y) = \begin{cases} 0 & \text{se } N_{STM} \geq 1, \\ I_{fo}, (x,y) & \text{se } N_{STM} < 1, \end{cases} \tag{3.32}$$

com $I_{fo} = \{I_{ff}, I_{of}\}$.

Este processamento tem como objectivo aumentar as regiões onde são detectados obstáculos de forma a tornar mais fácil a posterior consulta de informação. Mais uma vez perde-se al-

guma precisão que, como já foi referido, não é necessária. Por fim, processa-se a imagem do mapa de forma a que se entre duas regiões pretas houver uma única região branca, essa região também se torna preta, obtendo-se finalmente um esboço dos obstáculos que o robô encontra à sua frente, visível na Figura 3.14, em que à esquerda temos a imagem binária obtida na posição (P2,0H) e à direita o mapa criado a partir da mesma.

Figura 3.14: Construção do mapa da STM: à esquerda a imagem binária obtida I_{STM} na posição (P2,0H) e à direita o mapa criado a partir da mesma.

A LTM, por sua vez, consiste num mapa de maiores dimensões, pois tem o objectivo de auxiliar o robô na navegação global. Diferencia-se da STM em diversos aspectos tais como o seu tamanho (400×400 pixéis), o facto de o robô se poder localizar em qualquer zona do mapa, a resolução mais baixa do mesmo, e ainda a diferente forma de armazenamento da informação, não sendo um mapa binário. A LTM é construída a partir das imagens obtidas tanto na posição P1 como P2.

A construção deste mapa é feita através de sucessivos reforços positivos ou negativos: (a) se um obstáculo for detectado no mesmo local, recebe um reforço positivo a menos que tenha atingido um valor máximo M_R; (b) se um obstáculo já não se encontrar no mesmo local, recebe um reforço negativo. Para além destes reforços, existe ainda um reforço negativo associado ao tempo, que tem a função de provocar o esquecimento de informação pouco consistente. Este reforço (c) negativo é aplicado a cada intervalo de tempo, T, apenas aos obstáculos que não tenham atingido o valor máximo de reforço M_R, pois consideram-se que são obstáculos definitivos, por terem sido vistos muitas vezes no mesmo local.

Este método de reforços negativos e positivos permite construir um mapa em que a informação que foi registada um maior número de vezes tem um maior "grau de certeza"

do que outra que foi registada poucas vezes. Uma vez que as imagens obtidas utilizando P1 têm maior precisão do que as que são obtidas utilizando P2, o reforço (d) provocado pela informação extraída das primeiras tem também mais peso do que o reforço (e) provocado pelas segundas. Para fazer os diversos reforços na LTM primeiro é necessário criar um mapa temporário com o mesmo tamanho da LTM que reunirá a informação necessária de forma a indicar que pixéis devem ser reforçados positivamente e que pixéis devem ser reforçados negativamente. Esse mapa temporário inicialmente tem todos os pixéis a branco (255). Os pixéis que devem ser reforçados positivamente são depois colocados a preto (0) e os que devem ser reforçados negativamente são colocados a cinzento (127). Para criar esse mapa temporário e actualizar a LTM segue-se o seguinte processo:

1. Processam-se as imagens obtidas pelo robô de 5 em 5 pixéis, tanto horizontal como verticalmente, de forma a poupar tempo de processamento. Para cada um dos pixéis é calculada a posição relativa dos mesmos, em coordenadas polares, utilizando as expressões referidas anteriormente.

2. Convertem-se as coordenadas polares, $D.cis(\Theta)$, para coordenadas cartesianas, utilizando as seguintes equações:

$$c = \frac{N}{2} + sin(\Theta) \times D \times \frac{1}{2}, \tag{3.33}$$

$$l = M - cos(\Theta) \times D \times \frac{1}{2}. \tag{3.34}$$

   Ambas as equações são multiplicadas por $\frac{1}{2}$ para reduzir a resolução do mapa para metade, de modo a que cada pixel da imagem do mapa corresponda a $2cm$.

3. Tal como se faz para a STM, divide-se a imagem em regiões quadradas e não sobrepostas de tamanho $m \times m$. Se houver um pixel preto numa região, então a mesma torna-se preta. Caso não exista um pixel preto mas exista um pixel cinzento, então a mesma torna-se cinzenta.

   Este mapa provisório será então usado para actualizar a LTM. A LTM é também constituída por regiões $m \times m$, não sobrepostas. De forma a reduzir o tempo de processamento é apenas processado um pixel por cada uma dessas regiões, sendo registado o momento temporal, em segundos, em que foram actualizadas pela última vez.

4. Sempre que uma região não é actualizada há mais de um valor limite de T segundos (com $T = 300s$), essa região é reforçada negativamente de um valor N_R (utilizou-se $N_R = -20$), a menos que esse pixel já tenha atingido um valor de reforço máximo M_R (considerou-se $M_R = 200$), que define essa região como um obstáculo definitivo, ou a menos que esse pixel já não tenha qualquer reforço positivo.

5. Sempre que uma região é actualizada é registado o momento temporal dessa actualização. Esta primeira fase de reforços permite assim ao robô esquecer informação pouco consistente e ao mesmo tempo manter ou reforçar informação mais consistente ao longo do tempo.

6. Analisa-se o mapa provisório criado anteriormente, reforçando positivamente, com um valor P_R (utilizou-se $P_R = 40$), na LTM todos os pixéis que no mapa provisório se encontrem a preto e negativamente, com um valor N_R. Como referido anteriormente, as imagens captadas em P1 têm maior precisão que as captadas com P2, pelo que os reforços feitos a partir de P1 são sempre afectados de um valor k_R (usou-se $k_R = 3$). Deste modo os reforços feitos a partir de informação captada em P1 são $k \times P_R$, quando positivos e $k \times N_R$, quando negativos, com $k_R > 1$. Também nesta fase de reforços, é registado o momento temporal sempre que um pixel é actualizado.

   Na Fig. 3.15 pode visualizar-se à direita um exemplo de um mapa LTM construído pelo robô durante a exploração que consistia em percorrer o ambiente todo e depois retornar à base da *sandbox* apresentada à esquerda.

Figura 3.15: Exemplo de um mapa LTM criado pelo robô. À esquerda pode ver-se o ambiente de navegação do robô e à direita o mapa correspondente. Os objectos de referência encontrados estão marcados no mapa LTM sob a forma de quadrados de cores diferentes.

Para além da informação acerca dos obstáculos e limites encontrados, o robô guarda ainda uma lista de objectos de referência que vai encontrando no seu percurso. Estes objectos de referência têm que constar da biblioteca de objectos conhecidos do robô (a memória dos objectos faz parte da LTM). Cada vez que um desses objectos é encontrado é calculada a distância aproximada entre o robô e o objecto para determinar a sua localização. Para calcular a distância ao objecto, determinam-se os valores máximos e mínimos em x e y $(x_{min/max}, y_{min/max})$ da caixa que envolve os pontos-chave encontrados com correspondência com os pontos-chave da imagem do objecto existente na biblioteca de objectos. Este procedimento é equivalente ao que os humanos fazem, pois corresponde a fazer a segregação do objecto do fundo, para depois o comparar com o *template* que temos em memória [Rodrigues and du Buf, 2009a]. A seguir calcula-se o tamanho da diagonal D da caixa através da seguinte expressão:

$$D = \sqrt{(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2}. \tag{3.35}$$

Utilizando a mesma expressão calcula-se a diagonal D_t da caixa que envolve os mesmos pontos-chave, mas nos pontos-chave pré-existentes na base de dados, que se sabe à partida que foi tirada a uma distância de aproximadamente $45cm$ (corresponde à normalização, que também é um processo usado pelos humanos). Depois de se calcular o tamanho das duas diagonais, e sabendo que têm uma relação de proporcionalidade directa com a distância ao objecto, calcula-se o factor de escala que depois é multiplicado por $45cm$, obtendo-se finalmente uma distância aproximada, D_{ot} ao objecto detectado:

$$D_{ot} = \frac{D_t}{D_o} \times 45cm, \tag{3.36}$$

De forma a completar as coordenadas polares que permitam determinar a localização aproximada do objecto em relação ao robô calcula-se também um ângulo aproximado. Este ângulo obtém-se através da soma do ângulo de orientação do robô com o ângulo de rotação horizontal da cabeça do robô (ver Tabela 3.3). Por fim, tal como se fez para determinar a posição absoluta dos obstáculos no mapa, utilizam-se as equações 3.33 e 3.34.

Quando um objecto é localizado pela primeira vez a sua posição é registada, servindo a partir desse momento para que o robô possa corrigir a sua posição cada vez que o encontre. Quando o robô encontra um objecto cuja posição já tenha sido previamente determinada, o

robô calcula a nova distância ao mesmo, passando depois a considerar que a sua posição no mapa se encontra na mesma orientação, mas na nova posição correspondente a essa distância.

Para além desta correcção, o robô corrige também a sua orientação, desde que reencontre pelo menos dois objectos. Para tal é sempre registada a localização do último objecto encontrado. Quando é encontrado o segundo objecto calcula-se o ângulo feito entre os dois objectos, α_1. Seguidamente, em vez de se utilizar a posição do segundo objecto recentemente determinada, utiliza-se a posição do objecto determinada inicialmente e calcula-se o novo ângulo, α_2, que é considerado como o ângulo que está mais correcto, pois tem menos erro associado. Por fim, calcula-se a diferença entre os dois ângulos e soma-se essa diferença à orientação do robô.

Na Figura 3.16 estão exemplificadas as duas correcções que se podem fazer. À esquerda temos a correcção da posição por determinação da distância ao objecto. O robô, a azul, detecta um ponto de referência representado a cor de laranja à distância $D2$. Contudo, de acordo com a localização do objecto armazenada em memória, o objecto deveria estar à distância $D1$ (a vermelho). Para corrigir a sua posição o robô desloca as suas próprias coordenadas de $D1 - D2$ em direcção ao objecto. No caso da correcção da orientação, representada à direita na Fig. 3.16, o último objecto visto pelo robô (a verde) deveria fazer um ângulo α_1 com o ponto de referência encontrado. Contudo, de acordo com a posição em que o robô detectou o ponto de referência, esse ponto faz agora um ângulo α_2 com o último objecto encontrado. Para corrigir a sua orientação o robô soma à mesma a diferença entre α_1 e α_2, reduzindo o erro na sua orientação.

Utilizando os objectos como referências espaciais o robô pode assim compensar os erros de posicionamento e orientação que se vão acumulando ao longo da navegação devido a variados factores, como referido anteriormente. A eficiência da correcção da posição aumenta se aumentarmos o número de referências espaciais. Este processo duplica o comportamento dos humanos, uma vez que quando navegamos num ambiente vamos encontrando referências (portas, extintores, vasos, etc.) e utilizamos essas referências para nos localizarmos no espaço. Além disso, é também comum usarmos as duas/três mais próximas para corrigirmos a direcção. Se nos perguntarem onde está alguma dessas referências não conseguimos com precisão localizá-las, mas conseguimos dar uma orientação aproximada(e.g. encontra-se às três horas da posição actual) e uma distância aproximada (e.g. encontra-se a cerca de cinco metros). Neste caso, tendo em conta o tamanho do robô estamos a falar de valores na ordem

dos graus e dos centímetros.

Tal como acontece connosco, depois de nos afastarmos das referências e seguirmos um caminho desconhecido sabemos voltar ao ponto inicial mais tarde. Contudo, se nos perguntarem exactamente onde estão estas referências, nunca sabemos dizer com exactidão a sua posição, apenas a ordem pela qual se encontram ou a relação entre elas (que está mais próxima da outra, ou à direita, ou à esquerda, por exemplo).

O robô funciona da mesma forma, usando esta filosofia. Sabe a ordem das referências e a sua relação, mas não sabe com exactidão a sua posição, pois uma vez que não tem odometria não consegue colocar os objectos em posições exactas e pela mesma razão não consegue também localizar-se a si mesmo com exactidão. A auto-localização é feita em função das referências e linhas que limitam o ambiente.

Mais uma vez, pela observação do mapa LTM voltamos a salientar a falta de precisão que o robô tem em marcar (memorizar) as linhas que delimitam a *sandbox*. Realçamos que este é um comportamento típico humano que na robótica dita tradicional é intencionalmente corrigido, pois pretende precisão absoluta em todos os momentos.

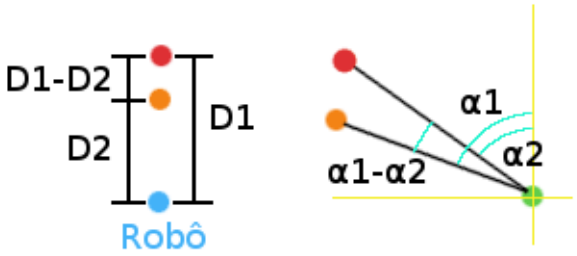

Figura 3.16: Diagramas representativos da correcção da posição e da orientação do robô.

Capítulo 4

Resultados

Resumo: Este capítulo faz a descrição dos testes efectuados para comprovar a funcionalidade da arquitectura proposta e apresentar os resultados dos mesmos.

4.1 Teste e Resultados

A fim de comprovar a funcionalidade da arquitectura cognitiva e do sistema de SLAM propostos realizaram-se uma série de cinco testes, tendo cada um uma complexidade superior ao anterior:

- **Teste 1:** Encontrar um objecto em ambiente desconhecido;
- **Teste 2:** Encontrar um objecto diferente em ambiente desconhecido;
- **Teste 3:** Encontrar um objecto em ambiente desconhecido e regressar ao início;
- **Teste 4:** Encontrar um objecto em ambiente desconhecido e regressar ao início mas encontrar um obstáculo (inexistente no momento da ida) no regresso;
- **Teste 5:** Encontrar um objecto em ambiente desconhecido, contar até 4 objectos conhecidos e regressar ao início;

Para efectuar os testes referidos utilizou-se a *sandbox* visível na Figura 3.3. Consiste numa área de aproximadamente $7m^2$ delimitada por fitas. Na *sandbox* dispuseram-se vários

objectos para serem reconhecidos pelo robô e utilizados como referência (ver Fig. 4.1 em cima, à direita).

Todo o processamento foi feito em tempo real num portátil com processador Intel Core 2 Duo 1.3GHz e com 4GB DDR3 de memória. Passamos a descrever cada um dos testes e respectivos resultados.

Encontrar um objecto num ambiente desconhecido (teste 1): este teste teve o objectivo de comprovar o funcionamento do bloco visio-motor referente à exploração e o sistema de localização e mapeamento simultâneos. Como referido na Secção 3.4.1, a exploração é feita com base em dois parâmetros principais: a saliência e a distância ao último objecto encontrado (ou à terceira última posição ocupada enquanto o robô não encontra o primeiro objecto de referência). Outra componente que se pretendeu testar foi o sistema visual nas suas diversas vertentes: com base na saliência conseguir extrair as zonas interessantes da imagem; processar cada uma das zonas em separado e por ordem da quantificação da saliência utilizada (neste caso, a média); por fim reconhecer objectos que possam existir dentro dessas zonas. Para efectuar este teste deu-se ao robô o comando simples de procurar uma caixa de ferramentas cinzenta e azul (ver Figura 4.1, à direita, na linha de cima). Dispuseram-se os objectos na *sandbox* do robô conforme o representado na Figura 4.1 à esquerda, na linha de cima. O robô começou na posição assinalada a amarelo, tendo assim que explorar e mapear o ambiente até encontrar o objecto pedido, que se encontra marcado a laranja (ver Fig. 4.1 em baixo, à esquerda).

O movimento do robô foi feito em passos de cerca de $25cm$ de cada vez e ao fim de cerca de 9 minutos e 21 segundos o objectivo foi cumprido. Na Figura 4.1 em baixo, à esquerda pode-se ver o mapa criado pelo robô durante a navegação, assim como o seu trajecto.

No que diz respeito à selecção de zonas de interesse nas imagens para posterior reconhecimento de objectos verificou-se que funciona como esperado, pois quando um dos objectos aparecia numa imagem captada normalmente era criada uma região contendo o objecto. Alguns exemplos são visíveis na Figura 4.1 em baixo, à direita.

Apesar do objectivo ter sido cumprido, foi notória a dificuldade em reconhecer a caixa de ferramentas cinzenta, pois só à terceira vez que o robô passou por ela é que foi reconhecida. No entanto os outros objectos foram reconhecidos com um número de correspondências significativo entre os pontos-chave das imagens pré-conhecidas e as imagens adquiridas. Verificou-se neste teste que o algoritmo de reconhecimento de objectos OpenSURF não funciona muito

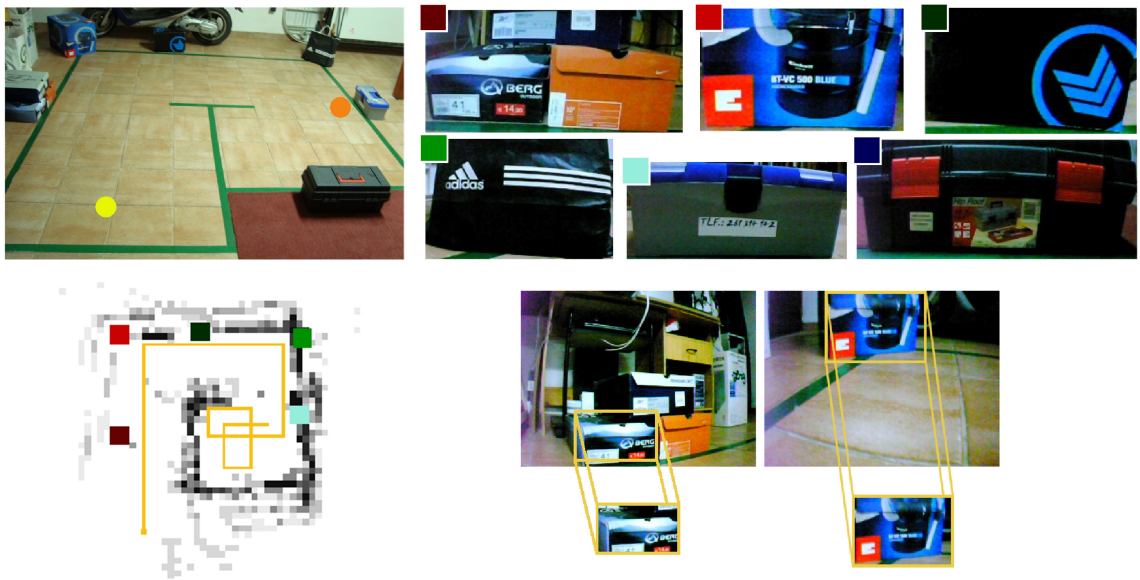

Figura 4.1: Em cima, à esquerda a disposição dos objectos no ambiente de teste para a primeira situação. Em cima, à direita, os objectos previamente inseridos na biblioteca de objectos conhecidos do robô. No canto superior esquerdo de cada imagem encontra-se a cor pela qual são representados nos mapas criados pelo robô. Em baixo, à esquerda, o mapa criado pelo robô durante a exploração até encntrar o objecto pretendido. O caminho percorrido pelo robô encontra-se traçado a cor de laranja. Cada um dos quadrados coloridos corresponde a um objecto reconhecido durante a tarefa. Em baixo, à direita, estão dois exemplos de zonas extraídas através da saliência que contém objectos, podendo-se efectuar o reconhecimento dos mesmos sem ser necessário processar a imagem completa.

bem para objectos com pouca textura.

Encontrar um objecto diferente em ambiente desconhecido (teste 2): Este teste é bastante semelhante ao primeiro, tendo como únicas diferenças a disposição dos objectos, o ponto inicial e o objecto que se pretende encontrar. Neste caso pretendeu-se comprovar que mesmo que as condições sejam diferentes a tarefa continua a ser realizada. Na Figura 4.2 à esquerda pode visualizar-se a nova disposição dos objectos, o novo ponto inicial e o novo objectivo. O mapa criado pelo robô durante a navegação está visível na Figura 4.2 à direita.

Mais uma vez a tarefa foi cumprida, tendo-se verificado o bom funcionamento do bloco visio-motor e do sistema visual implementado. Todavia, os problemas no reconhecimento das

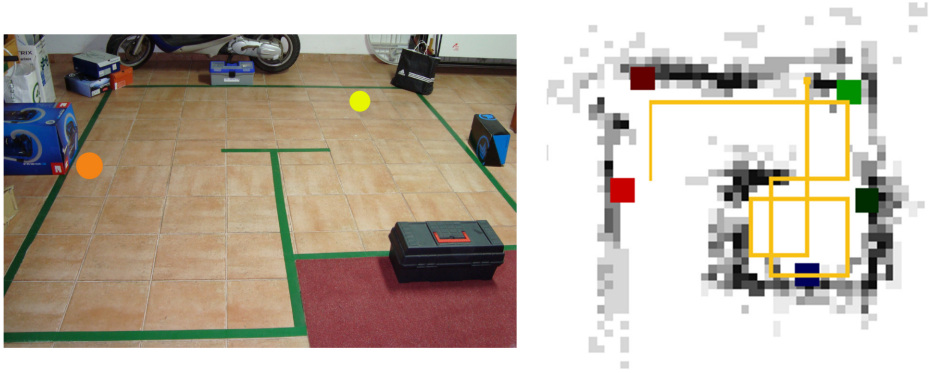

Figura 4.2: À esquerda a disposição dos objectos no ambiente de teste para a segunda situação. O ponto amarelo marca o local de início e o ponto laranja marca o objecto pelo qual o robô teve que procurar. À direita o mapa criado pelo robô durante a exploração até encontrar o objecto pretendido no segundo teste. O caminho percorrido pelo robô encontra-se traçado a cor de laranja. Cada um dos quadrados coloridos corresponde a um objecto reconhecido durante a tarefa.

caixas de ferramentas persistiram, revelando-se alguma dificuldade na detecção das mesmas. O tempo demorado foi semelhante ao do primeiro teste, pois foram necessários cerca de 10 minutos e 19 segundos para cumprir a tarefa.

Encontrar um objecto em ambiente desconhecido e regressar ao início (teste 3): com este teste pretendeu-se comprovar o funcionamento da componente de navegação em ambientes conhecidos. Foi comandado ao robô que encontrasse um determinado objecto e de seguida regressasse ao local inicial. Na Figura 4.3 à esquerda pode ver-se a disposição dos objectos no ambiente de testes, estando mais uma vez o local de início (e também de fim) marcado a amarelo e o objecto que se pretende encontrar a cor de laranja. À direita pode-se visualizar o mapa criado durante a navegação, assim como o trajecto do robô.

Esta tarefa foi bastante mais rápida que as anteriores, tendo levado cerca de 4 minutos e 8 segundos. Ao começar a fase de exploração da tarefa o robô reconheceu logo o objecto à sua esquerda, marcado a vermelho escuro. Alguns momentos depois reconheceu também o outro objecto marcado a vermelho e após mais algum tempo de navegação chega ao seu objectivo. Ao chegar ao seu objectivo, fez a consulta pelos objectos pelos quais tinha passado e fez o caminho inverso, passo a passo, movimentando-se sempre na direcção que encurtaria a distância entre si mesmo e o objecto seguinte. Através da saliência o robô pôde direccionar

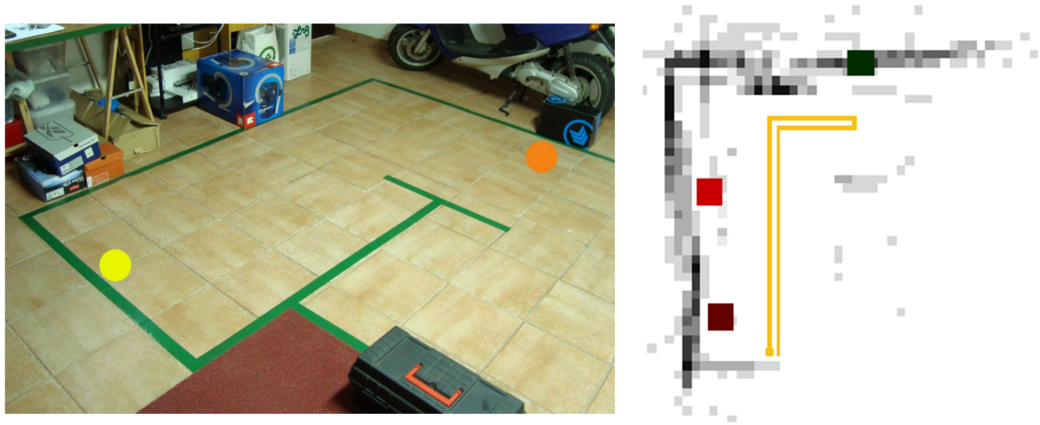

Figura 4.3: À esquerda a disposição dos objectos no ambiente de teste para a terceira situação. O ponto amarelo marca o local de início e fim e o ponto laranja marca o objecto pelo qual o robô teve que procurar. À direita o mapa criado pelo robô durante a exploração até encontrar o objecto pretendido no terceiro teste. O caminho percorrido pelo robô encontra-se traçado a cor de laranja. Cada um dos quadrados coloridos corresponde a um objecto reconhecido durante a tarefa.

a cabeça para as zonas mais salientes das imagens captadas, voltando assim a identificar os objectos previamente encontrados. Cada vez que um dos objectos era reencontrado, o robô iniciava o seu caminho até ao próximo objecto da lista. Ao chegar ao fim da lista de objectos percorridos durante a exploração, o robô estava novamente no local de início, concluindo assim a tarefa.

Com este teste verificou-se que uma vez conhecidos alguns locais do ambiente de navegação (no regresso à base, após a fase de exploração), o robô pode ir de objecto em objecto da mesma forma que nós, humanos, nos guiamos por pontos de interesse que vamos encontrando. Exemplo disso são as referências a objectos ou locais facilmente identificáveis quando damos indicações a outra pessoa de como chegar a algum local.

Teste 4: Encontrar um objecto em ambiente desconhecido e regressar ao início mas encontrar um obstáculo (inexistente no momento da ida) no regresso (teste4): a situação criada neste caso é bastante semelhante à anterior, tendo como única diferença o aparecimento de um obstáculo que não existia aquando da passagem por aquele lugar, aparecendo apenas no caminho de regresso, de forma a verificar a capacidade do robô em se adaptar a mudanças no ambiente. A disposição dos objectos foi a mesma da situação

anterior e pode ser vista na Fig. 4.3 à esquerda. Na Figura 4.4 à esquerda encontra-se o mapa construído pelo robô. Ao mapa foi adicionada a indicação de onde foi colocado o obstáculo. O obstáculo utilizado foi um x-acto cor de laranja, visível na Fig. 4.4 à direita.

Figura 4.4: À esquerda o mapa criado pelo robô durante a exploração até encontrar o objecto pretendido no terceiro teste. O caminho percorrido pelo robô encontra-se traçado a cor de laranja. Cada um dos quadrados coloridos corresponde a um objecto reconhecido durante a tarefa. Indicado a azul está o local onde se colocou o obstáculo. À direita pode ver-se o obstáculo colocado à frente do robô no caminho de regresso.

O robô repetiu o mesmo processo descrito na Secção anterior, exceptuando a situação em que aparece um obstáculo no seu caminho. Quando o obstáculo foi encontrado, o robô seguiu na direcção que resultaria no segundo maior encurtamento da distância entre ele e o obstáculo. Assim que o caminho ficou livre para voltar a seguir na direcção que seguia anteriormente o robô voltou a seguir nessa direcção. O aparecimento de um obstáculo foi facilmente resolvido, tendo a realização da tarefa demorado apenas 5 minutos e 2 segundos.

Encontrar um objecto em ambiente desconhecido, contar até 4 objectos conhecidos e regressar ao início (teste 5):

Neste último teste pretendeu-se comprovar o funcionamento do sistema de gestão de tarefas, sendo dados comandos ao robô para procurar um determinado objecto, encontrar um total de 4 objectos, e por fim, voltar ao início. Cada uma das tarefas tem o seu bloco visio-motor, sendo executado apenas o bloco visio-motor de uma tarefa de cada vez. No meio de cada um são executados sequencialmente os blocos de detecção e de verificação de cumprimento do objectivo das três tarefas. Deste modo, apesar de haver sempre uma tarefa principal a decorrer, a componente de detecção ds outras tarefas também vai sendo realizada.

Na Figura 4.5 à esquerda pode ver-se a disposição dos objectos e à direita está o mapa criado durante a realização das tarefas.

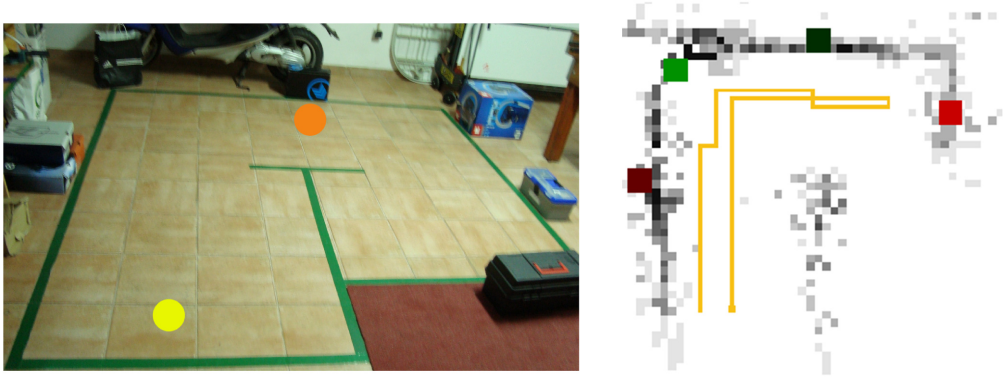

Figura 4.5: À esquerda a disposição dos objectos no ambiente de teste para a quinta situação. O ponto amarelo marca o local de início e de fim e o ponto laranja marca o objecto pelo qual o robô teve que procurar na realização da primeira tarefa. À direita o mapa criado pelo robô durante a a realização das três tarefas pretendidas no quinto teste. O caminho percorrido pelo robô encontra-se traçado a cor de laranja. Cada um dos quadrados coloridos corresponde a um objecto reconhecido durante a tarefa.

Este último teste foi também realizado com sucesso. O robô começou por procurar o objecto pedido, mas ao mesmo tempo cada vez que encontrava outro objecto somava um ao contador de objectos do bloco de detecção da segunda tarefa, que era contar até quatro objectos. Como tal, ao finalizar a primeira tarefa, durante a qual encontrou três objectos, continuou a busca pelo quarto objecto. Após completar a segunda tarefa, regressou ao início passando por todos os objectos que tinham sido encontrados durante a realização das duas primeiras tarefas.

Na Figura 4.6 pode visualizar-se um diagrama representativo da macro-tarefa realizada. Essa macro-tarefa é composta por três outras macro-tarefas, que para esta situação passaram a micro-tarefas para esta macro-tarefa, contendo cada uma delas três blocos: um visio-motor(azul), um de detecção(verde) e outro de verificação de conclusão da tarefa(laranja). Os blocos azuis são colocados num *buffer*, e os verdes e os laranjas noutros dois *buffers*. Estas macro-tarefas mais básicas têm que ser construídas pelo programador conjugando várias micro-tarefas de forma a criar o ciclo de acção pretendido.

O robô começa por executar o bloco visio-motor da primeira tarefa, que no caso da tarefa em questão é um bloco de acções visio-motoras direccionadas para a exploração. Em cada ciclo desse bloco o robô vai executar as tarefas presentes nos outros dois *buffers*, procedendo à detecção de objectos para ver se encontra o objecto que procura, à contagem de objectos reconhecidos e à detecção de objectos para posterior regresso ao início, que não tem qualquer efeito na tarefa porque o trajecto para regressar ao início ainda não está definido.

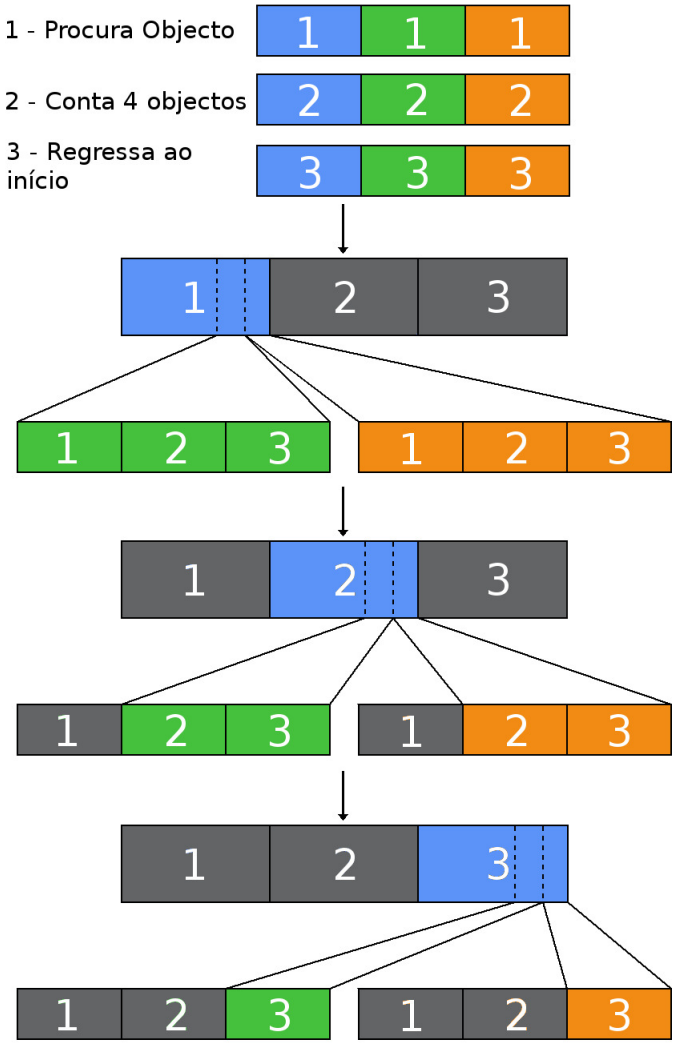

Figura 4.6: Diagrama representativo da macro-tarefa composta por 3 macro-tarefas mais simples.

Além das funções de detecção, são também executadas as funções que verificam se o objecto procurado já foi encontrado, se já se encontraram quatro objectos, e se já se voltou ao início, que não tem efeito porque a tarefa ainda não se iniciou. Quando o robô encontra o

objecto procurado termina a execução do bloco visio-motor correspondente e passa ao bloco visio-motor da tarefa 2, que é tambem um bloco direccionado para a exploração. Mais uma vez o robô executa os blocos verdes e laranjas correspondentes às restantes tarefas em cada ciclo de acção. Quando encontra o quarto objecto, dá por terminada a tarefa e inicia o bloco de acção visio-motora da tarefa de regresso ao início. Nesta tarefa é definido o caminho de regresso ao início e as funções de detecção e verificação começam a ter efeito. Como as duas primeiras tarefas já tinham sido realizadas, nesta última o robô ficou apenas a executar os blocos pertencentes à própria tarefa.

Com este conjunto de cinco situações foi possível verificar que a arquitectura para robôs apresentada nesta dissertação funciona como esperado, pois comprovou-se o funcionamento de cada uma das componentes implementadas.

4.2 Discussão

Depois de se terem efectuado os testes que comprovam o funcionamento da arquitectura cognitiva é importante situar esta arquitectura relativamente a outros dois desenvolvimentos que também foram referidos ao longo desta dissertação, os quais têm mais partes coincidentes com este trabalho e também a partir dos quais foram adquiridos alguns dos conceitos necessários no desenvolvimento da arquitectura descrita.

A proposta de Kawamura et al. [2002] para uma navegação egocêntrica é, sem dúvida, de entre os trabalhos referidos, o que tem o conceito mais semelhante ao que foi aqui apresentado. Como pontos comuns podem-se enunciar a utilização de uma estrutura de memória bipartida em memória a curto prazo e memória a longo prazo, a utilização da visão como meio de aquisição de informação do ambiente envolvente, a utilização de mapas sem métrica precisa, a capacidade de adaptação em ambientes dinâmicos e a clara inspiração na biologia. Contudo, existem também um conjunto de componentes na qual existe uma clara diferenciação entre os dois trabalhos:

(a) a arquitectura aqui apresentada inclui um sistema de localização e mapeamento simultâneo, permitindo-lhe navegar em ambientes desconhecidos e mapear esses ambientes, enquanto a proposta de Kawamura et al. [2002] requer a existência de um esboço do mapa *a priori* para que possa navegar e auto-localizar-se;

(b) a possibilidade de combinar tarefas para realizar tarefas mais complexas, sendo esse planeamento feito pelo robô com base nos conhecimentos prévios e de acordo com as instruções que lhe forem dadas. No trabalho de Kawamura et al. [2002] o operador define um conjunto de pontos com a trajectória aproximada com a finalidade do robô atingir um único objectivo;

(c) a inclusão de um sistema visual complexo, envolvendo Foco-de-Atenção, inibição de retorno e reconhecimento de objectos, que permite tornar a navegação e a aquisição de informação mais objectiva, enquanto que no trabalho de Kawamura et al. [2002] o sistema visual é bastante simples, consistindo apenas num sistema de reconhecimento de *tags* coloridas.

Outro trabalho que pode também ser considerado de grande importância no desenvolvimento da arquitectura referida ao longo desta dissertação foi o de Meger et al. [2008] que

tal como a arquitectura aqui apresentada também apresenta um sistema visual complexo, combinando saliência, reconhecimento de objectos e visão *stereo*. Além do sistema de visão complexo, na proposta de Meger et al. [2008] também é feito o mapeamento do ambiente, mas com um sensor laser em vez de ser feito com base na visão, ficando os mapas com melhor resolução e mais precisos, ao contrário do apresentado nesta dissertação. A navegação é feita com base em coordenadas métricas, sendo utilizado o algoritmo A* (*A-star*) para planear as rotas a seguir.

No capítulo seguinte vamos tirar as conclusões sobre o trabalho realizado e apresentar propostas de desenvolvimento futuro, pois terá que haver ainda muito trabalho futuro até se conseguir finalmente criar um robô completamente cognitivo.

Através das comparações efectuadas tornam-se evidentes os pontos em que o trabalho aqui descrito se aproxima e se diferencia de outros trabalhos em que foi baseado. Independentemente de ser mais completo ou incompleto que os outros trabalhos referidos em algumas componentes no que diz respeito à componente cognitiva, a grande certeza que se pode tirar é que terá que haver ainda muito trabalho futuro até se conseguir finalmente criar um robô completamente cognitivo.

Capítulo 5

Conclusão

Resumo: Este capítulo apresenta as conclusões obtidas relativamente ao trabalho desenvolvido. São ainda feitas algumas considerações relativamente ao trabalho futuro.

Esta dissertação apresenta uma proposta de uma arquitectura cognitiva para robôs que engloba um sistema visual, um sistema de memória e um sistema de realização de tarefas e aprendizagem. Além da arquitectura cognitiva é ainda apresentado um sistema de localização e mapeamento simultâneo. Como contribuições podem-se salientar:

(a) a implementação de um sistema visual que conjuga saliência, Foco-de-Atenção e reconhecimento de objectos;

(b) a implementação de um sistema de localização e mapeamento simultâneos directamente integrado com as estruturas de memória a curto e a longo prazo e que tem em conta o factor "tempo", permitindo assim a adaptação a ambientes dinâmicos e permitindo também eliminar progressivamente a informação com pouca coerência;

(c) a implementação de um sistema de gestão, construção de tarefas e aprendizagem com base em três tipos de blocos distintos.

O objectivo desta dissertação foi atingido, pois conseguiu-se criar um sistema que permite a exploração de um ambiente desconhecido, a navegação em ambientes conhecidos, a adaptação a mudanças no ambiente e a realização de tarefas, sendo todas estas capacidades

potenciadas por um sistema visual mais complexo que os sistemas visuais habitualmente utilizados e por uma estrutura de memória bipartida que permite efectuar a gestão da informação recolhida, o que também não é comum na robótica móvel usual.

No que respeita ao sistema visual, verificou-se que a combinação de saliência, o reconhecimento de objectos permite tornar a recolha de informação do meio ambiente mais objectiva, pois as zonas mais salientes são processadas em primeiro lugar e o posterior reconhecimento de objectos que é feito nessas zonas permite obter um tipo de informação bastante detalhado e versátil. Para além de tornar a recolha de informação mais objectiva, aumenta também a objectividade da navegação do robô, permitindo que este se mova para as zonas que à partida têm maior saliência. No entanto, pode-se considerar que o sistema visual poderia ainda beneficiar da implementação de visão *stereo*, de forma a ter-se também noções de profundidade no campo de visão.

Contudo, a implementação de funções interpoladoras para a determinação de distâncias permitiu atingir os objectivos propostos, pelo que a implementação de visão *stereo* pode ser considerada como trabalho futuro. Outra melhoria que poderia ser feita a nível do sistema visual seria a utilização de outro algoritmo de reconhecimento de objectos para objectos pouco texturados, pois verificaram-se dificuldades na detecção de objectos desse género por originarem baixas quantidades de pontos-chave. No futuro pretende-se que este algoritmo de reconhecimento de objectos seja substituído por um de reconhecimento de objectos completamente biológico [Rodrigues and du Buf, 2009b].

Quanto ao sistema de memória, conclui-se que a divisão em memórias a curto prazo e a longo prazo conjugadas com um factor de desvanecimento temporal permite uma gestão eficiente da informação, eliminando informação inconsistente e guardando apenas a informação relevante. O sistema é também eficiente do ponto de vista em que para uma navegação imediata basta recorrer à memória a curto prazo, sendo apenas necessário recorrer à utilização da memória a longo prazo quando é necessário planear uma acção.

O sistema de localização e mapeamento simultâneos proposto mostrou-se funcional, permitindo a navegação do robô mesmo em situações em que o ambiente varia. Contudo, para que este sistema seja mais preciso de forma a colmatar a inexistência de um sistema de odometria é necessária a utilização de vários pontos de referência de modo que o robô possa corrigir frequentemente a sua posição, que é o que o ser humano usualmente também faz. Se o número de objectos de referência for muito pequeno, as correcções à posição e orientação

raramente acontecem, não sendo possível criar um mapa muito fiável do ambiente devido à acumulação de erros.

A gestão e a construção de tarefas, apesar de terem um funcionamento relativamente simples também se mostraram funcionais, permitindo ao robô realizar as tarefas descritas na Secção 3.4. A variedade de tarefas que podem ser realizadas será tanto maior quanto maior for a quantidade de experiências passadas pelo que o desenvolvimento de mais blocos de construção de tarefas pode ser considerado como trabalho futuro.

Como conclusão final, pode dizer-se que o objectivo do trabalho apresentado nesta dissertação foi atingido mas há ainda muito a fazer na área da robótica cognitiva, pois qualquer uma das componentes apresentadas pode ainda ser alvo de inúmeras melhorias e podem ainda ser acrescentadas muitas outras componentes que permitam auxiliar e melhorar a componente cognitiva do robô.

5.1 Trabalho Futuro

Como foi evidenciado ao longo deste trabalho, a robótica cognitiva é uma área muito abrangente que envolve diversas componentes distintas. Deste modo é natural que existam muitas possibilidades no que diz respeito a novos desenvolvimentos e a melhoria das componentes já desenvolvidas. No entanto, o trabalho futuro pode passar pelos seguintes tópicos:

- Passar a fazer uso das duas câmaras para utilização de visão *stereo*, pois este tipo de visão permite obter informação mais detalhada sobre o meio envolvente. Este tipo de visão permitiria ainda deixar de utilizar a fita verde para definir os limites, uma vez que seria possível detectar os limites físicos do ambiente.

- Experimentar outros algoritmos de Foco-de-Atenção biológicos, tais como os referidos por Rodrigues and du Buf [2006].

- Experimentar o modelo cortical de Rodrigues and du Buf [2009a] para reconhecimento de objectos, que apesar de requerer grande capacidade de processamento pode dar origem a bons resultados, tendo em conta que é um modelo mais semelhante ao reconhecimento de objectos feito pelo ser humano.

- Tornar a aquisição e inserção de novos objectos na biblioteca de objectos conhecidos

automática, usando o conceito biológico de normalização e uma maior quantidade de vistas para reconhecer cada objectos.

- Desenvolver e optimizar uma biblioteca de tarefas, afim de aumentar as possibilidades de realização de tarefas por parte do robô. Melhorar também a relação micro- macro-tarefas.

- Fazer mais testes com um robô maior em ambientes reais (corredor, sala, etc.) de maneira a experimentar a arquitectura proposta neste trabalho utilizando a mesma perspectiva de visão que nós, humanos, temos.

Basicamente, o que se pretende no futuro é que além do robô ter parte dos processos baseados no ser humano, passe a ter todos os processos baseados no ser humano, incluindo todo o sistema visual e auditivo.

5.2 Lista de Publicações

A lista de publicações efectuadas durante o tempo de dissertação e mestrado são apresentadas de seguida. De referir que a publicação [1] não está directamente relacionada com o assunto desta dissertação, mas foram usados algoritmos dessa publicação neste trabalho, bem como nas publicações [3] e [4] tendo sido aplicados os algoritmos de reconhecimento de objectos usado nesta dissertação nesses trabalhos.

[1] Saleiro, M., Rodrigues, J. and du Buf, J.M.H. (2009) Automatic hand or head gesture interface for individuals with motor impairments, senior citizens and young children. Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI2009), Lisbon, Portugal, June 3-5, pp. 165-171.
[2] Saleiro, M., Rodrigues, J.M.F. and du Buf, J.M.H. (2010) Cognitive robotics: a new approach to simultaneous localisation and mapping, Proc. 16th Portuguese Conf. on Pattern Recogn. (RECPAD2010), Vila Real, Portugal, October 29, 2 pp.
[3] du Buf, J.M.H., Barroso, J., Rodrigues, J.M.F., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixiera, V., Saleiro, M. (2010) The SmartVision navigation prototype for the blind, Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), Oxford, United Kingdom, 25-26 November, pp. 167-174

[4] Saleiro, M., Rodrigues, J.M.F. and du Buf, J.M.H. (2011) A Bio-Inspired Approach to SLAM, in preparation fot IEEE 7th International Conference on Intelligent Computer Communication and Processing.

[5] du Buf, J.M.H., Barroso, J., Rodrigues, J.M.F., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixiera, V., Saleiro, M. (2011) The SmartVision Navigation Prototype for Blind Users, Accepted for International Journal of Digital Content Technology and its Applications.

Bibliografia

Al-Absi, H., Abdullah, A., 2009. A proposed biologically inspired model for object recognition. Proc. 1st Int. Visual Informatics Conference: Bridging research and practice 5857, 213–222.

Alami, R., Clodic, A., Montreuil, V., Sisbot, E. A., Chatila, R., 2006. Toward human-aware robot task planning. Association for the Advancement of Artificial Intelligence Spring Symposia, AAAI, 8pp.

Bar, M., Kassam, K., Ghuman, A., Boshyan, J., Schmid, A., Dale, A., Hämäläinen, M. S., Marinkovic, K., Schacter, D., Rosen, B., Halgren, E., 2006. Top-down facilitation of visual recognition. Proc. 1st Int. Visual Informatics Conference: Bridging research and practice 103 (2), 449–454.

Bay, H., Tuytelaars, T., Gool, L., 2008. Speeded-up robust features. Computer Vision and Image Understanding 110 (3), 346–359.

Brady, T., Konkle, T., Alvarez, G., Oliva, A., 2008. Visual long-term memory has a massive storage capacity for object details. Proc. Nat. Acad. Scie. Unit. Stat. Amer. 105 (38), 14325–14329.

Buschka, P., Saffioti, A., 1998. Some notes on the use of hybrid maps for mobile robots. Proc. 8th Int. Conf. Intelligent Autonomous Systems, 547–556.

Butko, N., Zhang, L., Cottrell, G., Movellan, J., 2008. Visual salience model for robot cameras. Proc. 2008 IEEE. Int. Conf. on Rob. and Auto. 2398–2403.

Butko, N. J., 2008. Nick's machine perception toolbox. `http://mplab.ucsd.edu/~nick/NMPT`.

Chelazzi, L., Miller, E. K., Duncan, J., Desimone, R., 2001. Respondes of neurons in macaque area V4 during memory-guided visual search. Cerebral Cortex 11 (8), 761–772.

Davison, A., Reid, I., Molton, N., Stasse, O., 2007. MonoSLAM: Real-time single camera SLAM. Proc. IEEE Int. Conf. on Robotics and Automation, 688–694.

Deco, G., Rolls, E. T., 2004. A neurodynamical cortical model of visual attention and invariant object recognition. Vision Res. 44 (6), 621–642.

Deco, G., Rolls, E. T., 2005. Attention, short-term memory, and action selection: A unifying theory. Prog. in Neurobiol. 76 (4), 236–256.

Dellaert, F., Stroupe, A. W., 2007. Linear 2D localization and mapping for single and multiple robot scenarios. Proc. IEEE Int. Conf. on Robotics and Automation 1 (6), 1052–1067.

du Buf, J., Barroso, J., Rodrigues, J., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixiera, V., Saleiro, M., 2010. The smartvision navigation prototype for the blind. Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion, 167–174.

Evans, C., January 2009. Notes on the opensurf library. Tech. Rep. CSTR-09-001, University of Bristol. URL http://www.chrisevansdev.com.

Forssén, P., Meger, D., Lai, K., Helmer, S., Little, J., Lowe, D., 2008. Informed visual search: Combinning attention and object recognition. IEEE Proc. Int. Conf. Rob. Aut. CFP08RAA-CDR (5), 935–942.

Gegenfurtner, K. R., Kiper, D. C., Levitt, J. B., 1997. Functional properties of neurons in macaque area V3. Journal of Neurophysiology 77 (4), 1906–1923.

Gonzalez, R., Woods, R., 2007. Digital Image Processing, 3rd Edition. Prentice Hall.

Grigorescu, C., Petkov, N., Westenberg, M. A., 2003. Contour detction based on nonclassical receptive field inhibition. IEEE Trans. Image Processing 12 (7), 729–739.

Hubel, D. H., 1995. Eye, brain, and vision, 2nd Edition. Scientific America Library.

Itti, L., Koch, C., 2001. Computational modeling of visual attention. Nature Reviews: Neuroscience 2 (3), 194–203.

Itti, L., Koch, C., Niebur, E., 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Patt. Recog. and Mach. Intel. 20 (11), 1254–1259.

José, J., Farrajota, M., Rodrigues, J., du Buf, J., 2010. A vision system for detecting paths and moving ostacles for the blind. Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion, 175–182.

Jung, Y., Choi, Y., Park, H., Shin, W., Myaeng, S.-H., 2007. Integrating robot task scripts with a cognitive architecture for cognitive human-robot interactions. Proc. IEEE Int. Conf. on Information Reuse and Integration, 152–157.

Kawamura, K., Koku, A., Wilkes, D., Peters II, R., Sekmen, A., 2002. Toward egocentric navigation. Int. J. Rob. Aut. 17 (4), 135–145.

Lee, T. S., 1996. Image representation using 2D gabor wavelets. IEEE Trans. Patt. Anal. Mach. Intel. 18 (10), 959–971.

Logothetis, N. K., Pauls, J., Poggio, T., 1995. Shape representation in the inferior temporal cortex of monkeys. Current Biology 5 (5), 552–563.

Lowe, D., 1999. Object recognition from local scale-invariant features. Int. Conf. Comp. Vis. 1150–1157.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. Int. Jour. Comp. Vis. 60 (2), 91–110.

Martins, J. A., Rodrigues, J., du Buf, J. M. H., 2008. Region segregation and saliency using colour information. Proc. 14th Port. Conf. Patt. Recog. 2pp.

Martins, J. A., Rodrigues, J., du Buf, J. M. H., 2009. Focus of attention and region segregation by low-level geometry. Proc. Int. Conf. Comp. Vis. Theo. App. 2, 267–272.

Masciocchi, C. M., Mihalas, S., Parkhurst, D., Niebur, E., 2009. Everyone knows what is interesting: Salient locations which should be fixated. Jour. Vision. 9 (11), 1–22.

Meger, D., Forssén, P., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J. J., Lowe, D. G., 2008. Curious George: An attentive semantic robot. Rob. Aut. Sys. 56 (6), 503–511.

Meinert, P., 2008. The impact of previous life experience on cognitive structure changes and knowledge acquisition of nursing theory and clinical skills in nontraditional nursing students. Doctoral Thesis, Kent State University, College of Education, Health and Human Services, United States of America, 173.

Miller, E. K., 2000. The prefrontal cortex and cognitive control. Nature Rev. Neuroscience 1 (1), 59–65.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. Fastslam: A factored solution to the simultaneous localization and mapping problem. Proc. AAAI. Nat. Conf. Art. Int. 593–598.

Nistér, D., Stewénius, H., 2006. Scalable recognition with a vocabulary tree. Proc. IEEE. Comp. Soc. Conf. on Comp. Vision and Patt. Rec. 2, 2161–2168.

Ohali, Y., 2011. Computer vision based date fruit grading system: design and implementation. Journal of King Saud Univ. - Comp. and Inf. Sciences (23), 29–36.

Olshausen, B. A., Field, D. J., 2005. Attention, short-term memory, and action selection: A unifying theory. Neural Computation 17 (8), 1665–1699.

Patnaik, S., 2007. Robot Cognition and Navigation: An Experiment with Mobile Robots, 1st Edition. Springer.

Qiu, F. T., von der Heydt, R., 2005. Figure and ground in the visual cortex: V2 combines stereoscopic cues with gestalt rules. Neuron 47 (1), 155–166.

Ramisa, A., Vasudevan, S., Scharamuzza, D., de Mántaras, R. L., 2008. A tale of two object recognition methods for mobile robots. Proc. 6th Int. Conf. Comp. Vision Systems, LNCS 5008, 353–362.

Rasche, C., 2005. The making of a neuromorphical visual system, 1st Edition. Springer.

Ratanaswasd, P., Gordon, S., Dodd, W., 2005. Cognitive control for robot task execution. Proc. IEEE Int. Work. Rob. Hum. Int. Com. (5), 440–445.

Rensink, R., 2000. The dynamic representation of scenes. Visual Cogn. 7 (1-3), 17–42.

Rodrigues, J., 2008. Integrated multi-scale architecture of the cortex with application to computer vision. Doctoral Thesis, University of the Algarve, Portugal, 156pp.

Rodrigues, J., du Buf, J. M. H., 2006. Multi-scale keypoints in V1 and beyond: Object segregation, scale selection, saliency maps and face detection. Biosystems 86 (1-3), 75–90.

Rodrigues, J., du Buf, J. M. H., 2009a. A cortical framework for invariant obect categorization and recognition. Cogn. Proc. 10 (3), 243–261.

Rodrigues, J., du Buf, J. M. H., 2009b. Multi-scale lines and edges in V1 and beyond: Brightness, object categorization and recognition, and consciousness. Biosystems 95 (3), 206–226.

Ruesch, J., Lopes, M., Bernardino, A., Hörnstein, J., Santos-Victor, J., Pfeifer, R., 2008. Multimodal saliency-based bottom-up attention. a framework fot he humanoid robot iCub. Proc. IEEE. Int. Conf. Rob. Aut. 962–967.

Saeedi, P., Lowe, D., Lawrence, P., 2003. 3D localisation and tracking in unknown environments. Proc. IEEE Int. Conf. on Robotics and Automation 1, 135–145.

Saleiro, M., Rodrigues, J., du Buf, J., 2009. Automatic hand or head gesture interface for individuals with motor impairments, senior citizens and young children. Proc. Int. Conf. Soft. Dev. for Enhancing Accessibility and Fighting Info-Exclusion, 165–171.

Saleiro, M., Rodrigues, J., du Buf, J., 2010. Cognitive robotics: a new approach to simultaneous localisation and mapping. Proc. 16th Portuguese Conf. on Pattern Recognition, 2pp.

Se, S., Lowe, D., Little, J., 2001. Vision-based mobile robot localization and mapping using scale-invariant features. Proc. IEEE. Int. Conf. Rob. Aut. 2, 2051–58.

Shotton, J., Blake, A., Cipolla, R., 2008. Efficiently combining contour and texture cues for object recognition. British Machine Vision Conference.

Smith, R., Lane, P. C. R., Gobet, F., 2008. Modelling the relationship between visual short-term memory capacity and recall ability. Proc. Euro. Symp. Comp. Mode. Sim. 99–104.

Thrun, S., Burgard, W., Fox, W., 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. Proc. IEEE Int. Conf. on Robotics and Automation 1, 321–328.

Thrun, S., Gutmann, J., Fox, D., Burgard, W., Kuipers, B., 1998. Integrating topological and metric maps for mobile robot navigation: A statistical approach. Proc. 15th Conf. on Artificial Intelligence 1, 989–995.

Tomatis, N., Nourkbakhsh, I. Siegwart, R., 2001. Simultaneous localization and map building: A global topological model with local metric maps. Proc. IEEE/RSJ. Int. Conf. Int. Rob. Sys. 1, 421–426.

Vasudevan, S., Gachter, S., Siegwart, R., 2006. Cognitive spatial representations for mobile robots - perspectives from a user study. Technical Report 165068-2006-01.

Weldon, P. W., Higgins, W. F., Dunn, D. F., 1996. Gabor filter design for multiple texture segmentation. Optical Engineering 35 (10), 2852–2863.

Wong, Y., 2005. A study of approaches for object recognition. Master of Philosophy Term Paper, University of Hong Kong, China, 23.

Zarowski, C. J., 2004. An introduction to numerical analysis for electrical and computer engineers. Wiley.

Zetzsche, C., Wolter, J., Schill, K., 2008. Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. Cognitive Processing 9 (4), 283–297.